

TARTU ÜLIKOO
LOODUS- JA TEHNOLOOGIATEADUSKOND
FÜÜSIKA INSTITUUT
FÜÜSIKA ÕPPEKAVA

Mattias Rennel

**KIIRGUSE KASUTAMISE EFEKTIIVSUSE PÕHIMÕTTEL TÖÖTAVA
MAISMAA TAIMKATTE PRIMAARPRODUKTSIOONI MUDELI
RAKENDUS PYTHON'IS**

Magistritöö (30 EAP)

Juhendaja: Piia Post (Tartu Ülikool, füüsika instituut)

Kiirguse kasutamise efektiivsuse põhimõttel töötava maismaa taimkatte primaarproduktsooni mudeli rakendus Python'is.

Lühikokkuvõte:

Kiirguse kasutamise efektiivsuse põhimõttel töötavad maismaa taimkatte primaarproduktsooni mudelid on viimasel kümnendil suuremat tähelepanu leidnud, sest nende rakendamine globaalseteks analüüsideks on sisendandmete olemasolu, suhteliselt lihtsa mudeli ülesehituse ja suhteliselt tagasihoidlike arvutusressursside tõttu võimalikuks saanud. Primaarproduktsooni mudelid aitavad analüüsida ja paremini mõista süsinikuringet, modelleerida ökosüsteemide käitumist ja hinnata taimestiku kasvu. MODIS GPP/NPP produkt on üks enim analüüsitud ühe sellist tüüpi mudeli tulem. MODIS'e mudeli põhjal on Eesti alale loodud mudeli EST-PP ning selle edasiarendus ja Python rakendus EST-PP-Python.

Käesolev töö tutvustabki lähemalt EST-PP-Python mudelit ja rakendust andes eelnevalt ka lähema ülevaate süsinikuringest, primaarproduktsoonist ja kiirguse kasutamise efektiivsuse põhimõttel töötavatest mudelitest.

Võtmesõnad:

Primaarproduktsoon, LUE, GPP, NPP, EST-PP.

CERCS:

P160 - Statistika, operatsioonanalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

P500 - Geofüüsika, füüsikaline okeanograafia, meteoroloogia.

P510 - Füüsiline geograafia, geomorfoloogia, mullateadus, kartograafia, klimatoloogia.

Light-use efficiency type terrestrial vegetation primary production model's implementation in Python.

Abstract:

Light use efficiency type terrestrial vegetation primary production models have found increasing attention in the last decade because their global application has become possible due to the emergence of relevant global input datasets, a relatively simple overall model structure and the relatively small amount of computing power needed to run them. Models of primary production help to analyse and better understand the carbon cycle, to simulate ecosystems

evolution and assess vegetation growth. The MODIS GPP/NPP product is one of the most analysed outputs of one light use efficiency type of models. Based on the MODIS model, EST-PP and its further development in Python, EST-PP-Python, are also such type models that have been applied for studying primary production in Estonia.

This work introduces the EST-PP-Python model and implementation with prior general introductions to the carbon cycle, primary production and light use efficiency type primary production models.

Keywords:

Primary production, LUE, GPP, NPP, EST-PP.

CERCS:

P160 - Statistics, operation research, programming, actuarial mathematics.

P500 - Geophysics, physical oceanography, meteorology.

P510 - Physical geography, geomorphology, pedology, cartography, climatology.

Sisukord

Tähiste loetelu.....	6
Sissejuhatus	7
1. Üldine teoreetiline taust.....	9
1.1. Süsinikuringe.....	9
1.2. Primaarproduktsoon	10
1.3. Primaarproduktsooni hindamine	11
2. KKE tüüpi primaarproduktsooni mudelid	13
2.1. Mudelist üldiselt	13
2.2. MODIS GPP/NPP	13
2.3. EST-PP	17
2.4. Peamised erinevused mudelite vahel.....	19
3. EST-PP-Python.....	22
3.1. Taust	22
3.2. Arenduse põhimõtted	22
3.3. Arenduse keskkond	23
3.4. Arenduse ülesehitus.....	23
3.5. Kasutamine	24
3.6. Tööpõhimõte ja funktsionaalsus.....	26
3.6.1. estpp.py	26
3.6.2. config.py.....	27
3.6.3. gpp.py	28
3.6.4. npp.py	30
3.6.5. outputs_processing.py	31
3.6.6. utils.py	32
3.6.7. outputs_analysis.py	32
3.7. Arenduse seis ja võimalikud edasiarenduse suunad.....	33

3.8. Arutelu.....	33
Kokkuvõte	35
Kasutatud kirjandus	36
Lisad	40
Lisa 1. user_config.yaml	40
Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	41

Tähiste loetelu

APAR – neeldunud fotosünteetiliselt aktiivne kiirgus (ingl. k. *absorbed photosynthetically active radiation*)

fAPAR – neeldunud ja pealelanguvad fotosünteetiliselt aktiivse kiirguse suhe (ingl. k. *fraction of absorbed photosynthetically active radiation*)

FPAR - neeldunud ja pealelanguvad fotosünteetiliselt aktiivse kiirguse suhe (ingl. k. *fraction of absorbed photosynthetically active radiation*)

GPP – brutoprimaarproduktioon (ingl. k. *gross primary production*)

GR – kasvuhingamine (ingl. k. *growth respiration*)

IPAR – pealelanguvad fotosünteetiliselt aktiivne kiirgus (ingl. k. *incident photosynthetically active radiation*)

KKE – kiirguse kasutamise efektiivsus

LAI – lehepinnaindeks (ingl. k. *leaf area index*)

LSWI – maapinna vee indeks (ingl. k. *land surface water index*)

NPP – netoprimaarproduktioon (ingl. k. *net primary production*)

MODIS – mõõduka ruumilise lahutusega spektraalradiomeeter (ingl. k. *moderate-resolution imaging spectroradiometer*)

MR – raku hingamine (ingl. k. *maintenance respiration*)

NEP – neto ökosüsteemi produktioon (ingl. k. *net ecosystem production*)

PAR – fotosünteetiliselt aktiivne kiirgus (ingl. k. *photosynthetically active radiation*)

SLA – erilehepindala (ingl. k. *specific leaf area*)

SM – maapinna niiskussisaldus (ingl. k. *soil moisture*)

VPD – veeaururõhu defitsiit (ingl. k. *water vapour pressure deficit*)

Sissejuhatus

Globaalne ja regionaalne süsinikuringe on populaarne uurimisvaldkond, sest nii süsiniku reservuaaride mahtudel kui ka nende vahelistel voogudel on suur mõju inimkonnale – kas siis inimtekkeliste kasvuhoonegaaside mõju hindamiseks erinevatele ökosüsteemidele [1] [2] [3] või rakenduslikuma poole pealt taimestiku, näiteks metsa või põllukultuuride, biomassi juurdekasvu analüüsimiseks [4]. Üheks süsinikuringe osaks on CO₂ neelamine maismaa taimkatte poolt ehk primaarproduktatsioon (või algproduktatsioon või esmasproduktatsioon), millele selles töös ka keskendutakse. Punktmõõtmiste abil on suhteliselt täpselt võimalik hinnata erinevaid ainevoogusid, nende hulgas ka eelnevalt mainitud CO₂ oma. Suure hulga kohtmõõtmiste arvu vajaduse tõttu ei ole selline meetod rakendatav usaldusväärsete regionaalsete ja globaalsete hinnangute saamiseks. Selleks on loodud ja välja pakutud erinevaid tüüpe mudeleid, nendest üheks on kiirguse kasutamise efektiivsuse (KKE) põhimõttel töötavad mudelid. Need on viimaste aastate jooksul palju tähelepanu saanud kuna on oma olemuselt suhteliselt lihtsad ja ka vajalikud sisendandmed globaalseteks ja regionaalseteks analüüsideks on üldjuhul kättesaadavad. Kuigi üles ehitatud samal põhimõttel, võib kirjandusest leida mitmeid erinevaid mudeli rakendusi, mis on motiveeritud olemasolevatest sisendandmetest või mudeli rakenduspiirkonna regionaalsetest isepäradest [5] [6] [7] [8] [9] [10]. Neid erinevaid rakendusi on viimasel ajal nii kohtmõõtmistega kui ka omavahel palju võrreldud, et mõista igaühe tugevusi ja nõrkusi ning leida regionaalselt kui ka globaalselt kõige paremini töötavad lahendused [1] [2] [3] [11] [12]. Nii saab hinnata ka taimkatte primaarproduktiooni mõjutavaid suurusid (näiteks temperatuur, maapinna niiskusesisaldus) kui ka nende regionaalseid isepärasusi.

Magistritöö eesmärk oli rakenduslik: ühe KKE tüüpi mudeli loomine programmeerimiskeeles Python, mida nimetame edaspidi EST-PP-Python, ja sellele esialgse manuaali tekitamine, mille ülesannet täidab see dokument. Aluseks said võetud üks Eesti alal rakendatud KKE tüüpi mudel EST-PP [13] ja selles kasutatud sisendandmestikud, millede järgi sai ka uut rakendust kohandatud ning testitud. Uut rakendust oli vaja sest EST-PP tulemused said loodud üksikute eraldiseisvate skriptide abil, mistõttu selle mudeli kasutamine ei olnud kasutajasõbralik ja nõudis palju aega ja tööd.

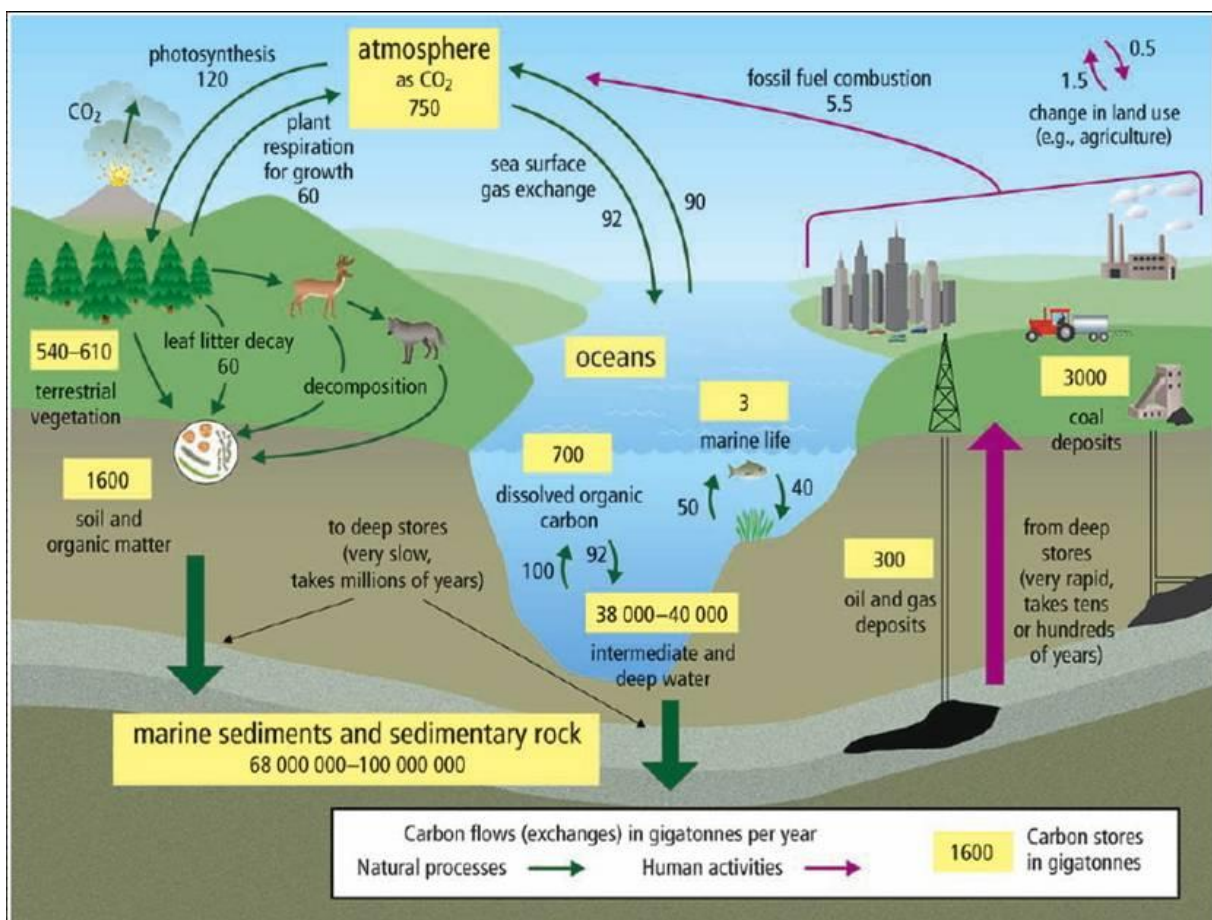
Töö põhitekst koosneb kolmest osast, neist kaks esimest on teoreetilist laadi ja kolmas kirjeldab rakendusliku ülesande lahendamist. Esimeses osas antakse lühike ülevaade süsinikuringest, primaarproduktatsioonist ja nende kirjeldamiseks kasutatavatest mudelitest. Teine osa keskendub sügavamalt KKE tüüpi primaarproduktiooni mudelite kirjeldamisele ja eraldi tuuakse

näidetena välja kaks sellist: MODIS'e primaarproduktiooni produktid ja EST-PP. Kolmandas osas tutvustatakse loodud EST-PP-Python mudelit, selle arendamise üldisi põhimõtteid, lõplikku rakendamist ja funktsionaalsust.

1. Üldine teoreetiline taust

1.1. Süsinikuringe

Süsinikuringe kirjeldab süsiniku liikumist ühes ökosüsteemis läbi selle ökosüsteemi erinevate reservuaaride (komponentide). Need komponendid võivad olla anorgaanilist laadi nagu atmosfäär või ookeanid, aga ka orgaanilist laadi nagu maapealne taimestik. Süsinikuringe kirjeldamisel on peamised kasutatavad kvantitatiivsed suurused süsiniku hulk nendes erinevates reservuaarides ja süsiniku vood nende reservuaaride vahel. Globaalset süsinikuringet läbi selle suuremate reservuaaride ja voogude kirjeldab joonis 1 [14].



Joonis 1. Globaalne süsinikuringe [14]. Nooltega on tähistatud süsiniku vood (rohelised jooned väljendavad looduslikke, punased inimtegevusest tingitud voogusid) ning nende suurused gigatonnides aastas. Kollasel taustal numbrid kirjeldavad vastava süsiniku reservuaari mahtu gigatonnides.

Maapealne taimestik pole, võrreldes teiste põhiliste reservuaaridega, just üks suurimaid komponente seal leiduva süsiniku hulga järgi. Küll aga on teda hõlmavad vood, nii bruto kui ka neto, suhteliselt suured ja ta on suurim atmosfääris leiduva süsiniku siduja. See teeb taimestiku uurimise oluliseks, kui tahta analüüsida inimtegevusest tingitud atmosfääri

lisandunud süsiniku (suuremas osas süsihappegaasi koostises) mõjusid aineringlusele ja ka üldiselt Maa ökosüsteemile.

Maapealne taimestik seob atmosfäärist süsinikku peamiselt läbi fotosünteesi ehk siis süsihappegaasi lagundamisel veeks ja orgaanilisteks molekulideks, päikeseenergia kaasabil. Umbes pool sellest süsinikust liigub taimede elutegevuse tõttu atmosfääri tagasi [5]. Ülejäänud seotakse enda kudedesse, kust see läbi taimetoiduliste elusolendite või maha langemise ja bakterite ning muude mikroorganismide poolt lagundamise kaudu, maapinda, ühte teise süsinikuringe komponenti, jõuab.

Teine peamine atmosfääris sisalduva süsiniku siduja on ookean ja seal leiduvad elusorganismid. Oma süsiniku mahutavuselt on ookean maapealsest taimestikust suurusjärke suurem reservuaar, kuid süsiniku vood tema ja atmosfääri vahel on samas suurusjärgus maapealse taimestikuga.

Atmosfäär, maapealne taimestik ja elusorganismid, ookean koos seal leiduvate elusorganismidega ning vood kõigi nende komponentide vahel moodustavad niinimetatud kiire süsinikuringe [15]. Kiiret süsinikuringet võib kirjeldada ka kui süsiniku liikumist biosfääris. Inimtekkelise kliimasoojenemise vaatenurgast on just see enim huvi pakkuv süsinikuringe osa, sest vood teiste ringe komponentide vahel on palju stabiilsemad ja väiksemad ning seetõttu palju vähem atmosfääri lisandunud süsiniku hulgast mõjutatud.

1.2. Primaarproduktsoon

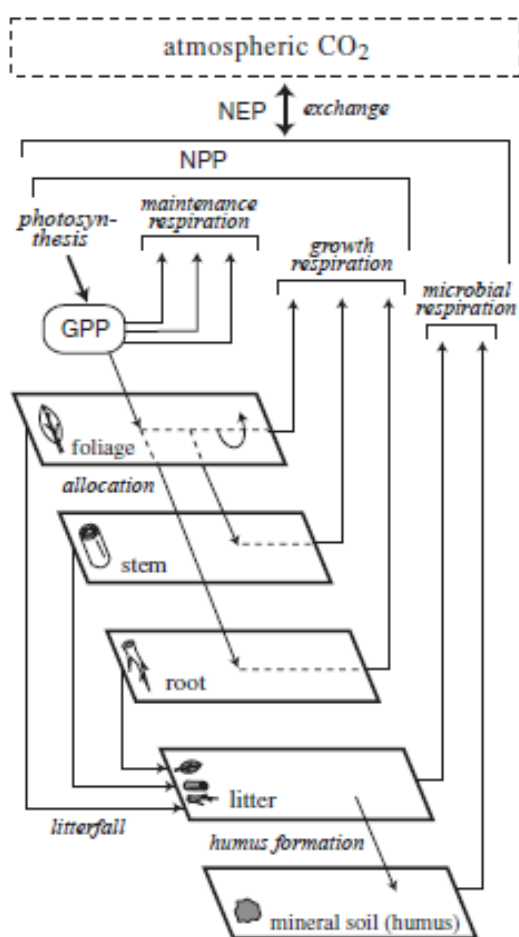
Primaarproduktsoon ehk algtoodang ehk esmane toodang (esmastoodang) on orgaaniliste ühendite valmistamine süsihappegaasi abil foto- või kemotsünteesi kaudu [16]. Seega on see ka põhiline protsess, mille kaudu käib süsiniku vahetus atmosfääri ja biosfääri eluskomponentide vahel. Organisme, mis sellest protsessist osa võtavad, nimetatakse autotroofideks ehk organismideks, kes enda kudede ehituseks vajamineva süsiniku hangivad anorgaanilistest süsinikuühenditest. Heterotroofid, vastupidiselt autotroofidele, hangivad vajamineva süsiniku orgaanilistest ühenditest ehk siis peamiselt kas taimedest või teistest elusolenditest. Primaarproduktsoon on enamiku eluslooduse eksisteerimise aluseks, kuna selle tootjad moodustavad toiduahela aluse. Maapinnal on algtootjateks enamjaolt fotosünteesivad taimed, ookeanides aga vetikad.

Primaarproduktsooni kirjeldamisel on kasutuses kaks peamist suurust: bruto- ja netoproduktsoon. Brutoproduktsoon (GPP, inglise keeles *Gross Primary Production*) on algtootjate poolt otseselt neelatud süsiniku hulk. Osa sellest süsinikust jõuab aga atmosfääri

tagasi organismide elutegevuseks vajaminevate protsesside tõttu, nagu näiteks raku hingamine taimedes, mis on biokeemilise energia tootmine eelnevalt fotosünteesi abil toodetud orgaanilistest ühenditest. Netoproduktioon (NPP, inglise keeles *Net Primary Production*) on brutoproduktiooni ja sellist tüüpi protsesside poolt atmosfääri tagasi jõudva süsiniku hulga vahe ehk teisisõnu ka organismide poolt pikemaks ajaks seotud süsiniku hulk. Taimkatte bruto- ja netoproduktiooni seob seega valem

$$\text{GPP} = \text{NPP} + R_a, \quad (1)$$

Kus R_a tähistab autotroofide erinevate hingamisprotsesside (näiteks raku hingamine ehk sisehingamine) poolt vabastatud (atmosfääri tagasi paisatud) süsiniku hulka.



Ökosüsteemide süsinikuringe seisukohalt on tähtsad suurused veel ka neto ökosüsteemi produktioon, mis lahutab netoproduktioonist maha heterotroofide poolt atmosfääri vabastatud süsiniku ja neto bioomi produktioon, mis võtab arvesse ka süsiniku kaod looduslikest või inimtekkelistest häiritustest (näiteks metsaraie või -tulekahju).

Antud töös keskendutakse maismaa taimkatte bruto- ja netoproduktioonile. Erinevaid produktiooni suurusi ja täpsemalt süsiniku ringlust metsades läbi selle erinevate komponentide kirjeldab ka joonis 2.

Joonis 2 (vasakul) [17]. Metsakoosluse süsinikuringe läbi tema erinevate komponentide.

1.3. Primaarproduktiooni hindamine

Primaarproduktiooni hindamiseks on mitmeid võimalusi – seda on võimalik hinnata nii kohtmõõtmistega kui ka vastavate mudelite väljatöötamise ja rakendamisega [3]. Kohtmõõtmistest on enimlevinud lahendus mastide või tornide paigaldamine koos erinevate meteoroloogilisi suuruseid ja õhu koostist analüüsivate instrumentidega. Ka kambermõõtmisi

kasutatakse ainevoogude mõõtmiseks, aga nende tulemuste laiendamine ökosüsteemi tasandile on keeruline.

Mudelid võib üldistades jagada kahte gruppi: protsessipõhised mudelid, mis kirjeldavad taimestikus aset leidvaid bioloogilisi ja biokeemilisi protsesse ning mingite üldisemate mõõtmiste põhjal loodud statistilisi või kontseptuaalseid mudeleid, mis üritavad primaarproduktiooni siduda mõne üldisema mõõdetava suurusega [4] [7] [17]. Metoodika valik sõltub eelkõige analüüsitavast ökosüsteemist ja selle ulatusest ning sisendandmete olemasolust. Lokaalsete hinnangute saamiseks on parim lahendus kasutada otseseid mõõtmisi või kui mudeldada, siis protsessipõhiseid mudeleid või mõnede kohapeal käsitsimõõdetavatel suurustel põhinevaid statistilisi mudeleid [4]. Protsessipõhised mudelid võivad anda päris täpseid hinnanguid, lisaks on nendega võimalik hinnata ka süsiniku liikumist läbi ökosüsteemi sisemiste komponentide. Nende miinuseks on suur vajalike sisendite arv, ilma milleta ei ole lihtsalt võimalik adekvaatselt taimede füsioloogilisi protsesse modelleerida.

Metsade juurdekasvu hindamisel on pikk ajalugu, tingituna selle teadmise praktilistest vajadusest metsatööstuses. Selleks on välja töötatud mitmeid allomeetrilisi võrrandeid, mis seovad mõned puudelt või puistust mõõdetavad suurused, näiteks rinnasdiameeter ja puude arv hektaril, metsa biomassi ja kordusmõõtmiste olemasolul juurdekasvuga. Globaalsete hinnangute saamiseks ei ole sellised meetodid rakendatavad kuna vajaksid suurel hulgal regulaarseid välitõid, mis kasvõi regionaalsel/riiklikul tasemel on juba suur töö [4]. Globaalsete hinnangute saamiseks on kasutatud leidnud KKE tüüpi mudelid. Seda seetõttu, et need põhinevad suhteliselt lihtsal algprintsibil, et mingi teatud taimekoosluse annab neelatud fotosünteesiliselt aktiivse kiirguse energiaühiku kohta kindla maksimaalse algtoodangu (siit ka nimi – kiirguse kasutamise efektiivsuse tüüpi mudelid) [18] [19], mida piiravad erinevad meteoroloogilised või kasvukohatüübist tulenevad keskkonnavalused faktorid. Selliseid mudeleid ongi enim rakendatud globaalsete algtoodangu hinnangute saamiseks, sest vajaminevaid sisendandmeid on suhteliselt vähe ja eksisteerivad neid kirjeldavad globaalsed andmekihid, millest mitmeid on võimalik luua satelliitmõõtmistest. KKE tüüpi mudelid muudaks veelgi atraktiivsemaks võimalus neid rakendada ka usaldusväärsete regionaalsete hinnangute saamiseks, mida on siiani pärssinud satelliidimõõtmiste tulemite madal ruumiline või ajaline lahutus.

2. KKE tüüpi primaarproduktiooni mudelid

2.1. Mudelitest üldiselt

Nagu juba mainitud, on KKE tüüpi mudelite puhul tegemist taimkatte primaarproduktiooni hindamisega läbi fotosünteesilise kiirguse neelamise, selle kasutamise efektiivsuse ja seda efektiivsust piiravate nähtuste kirjeldamise. Selle võib kokku võtta järgnevaks üldiseks võrrandiks [11]

$$\text{GPP} = \text{PAR} \times f\text{APAR} \times \epsilon_{\max} \times f(T, W, \dots), \text{ kus} \quad (2)$$

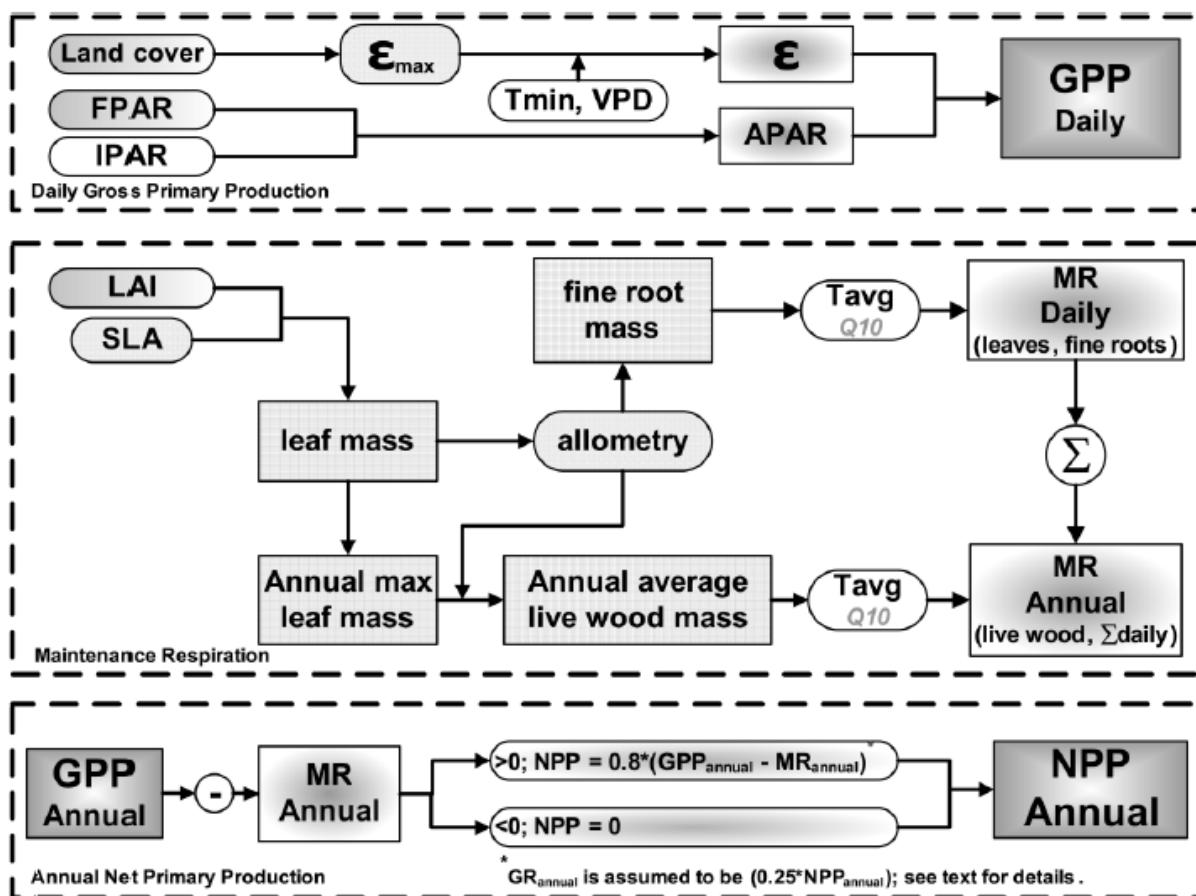
GPP = brutoproduktioon ($\text{MJ} \cdot \text{m}^{-2}$), PAR = fotosünteesiliselt aktiivse pealelangenu kiirguse hulk (MJ), fAPAR = taimestiku poolt neelatud fotosünteesiliselt aktiivse kiirguse suhe pealelangenu kiirgusesse, ϵ_{\max} = maksimaalne kiirguse kasutamise efektiivsus ($\text{gC} \cdot \text{m}^{-2} \cdot \text{MJ}^{-1}$) ja $f(T, W, \dots)$ = erinevaid piirangufunktsioone ühildav funktsioon. Sama tüüpivalemiga on üritatud ka netoproduktiooni hinnata [18], aga enamjaolt kasutatakse sellist lähenemist ikkagi brutoproduktiooni puhul. Sellest omakorda leitakse netoproduktioon rakuhingamise arvesse võtmisega, mille hindamiseks kasutatakse tihti peale taimefüsioloogiast tuletatud hingamist kirjeldavaid valemite [5]. Toodud valemis (2) on PAR ja tihti peale $f(T, W, \dots)$ leitavad globaalsete või regionaalsete ilmapudelite või nende reanalüüside väljunditest. fAPAR hinnatakse üldiselt satelliidimõõtmistest. Enim peavalu tekitab ϵ_{\max} hindamine, mis üldjuhul tuletatakse kaudselt mingitest mõõtmistest või kirjanduses leiduvatest hinnangutest.

2.2. MODIS GPP/NPP

Järgnevalt tutvustame kahte GPP ja NPP mudelit täpsemalt, et kirjeldada detailsemalt nende toimimist ja ülesehitust, ning selle peatüki viimases alampeatükis saab kirjeldatud lühidalt ka ülejäänuid teaduskirjanduses enim kajastust leidnud KKE tüüpi mudeleid, nende erinevusi ja omapärasid.

MODIS'e GPP ja NPP produktid [5] on ühed enim kasutatud ja analüüsitud primaarproduktiooni tulemid [20]. Seda seetõttu, et 10 aastat tagasi olid need ühed esimesed globaalse kattuvusega sellist tüüpi andmekomplektid. Selle tegi võimalikuks tehnoloogia areng, peamiselt arvutusressurssides, ja erinevate vajalike globaalsete sisendandmekomplektide olemasolu, mitmed neist on toodetud instrumendi MODIS kogutud andmete pealt sama NASA programmi raames, milles ka GPP/NPP produkt loodi. Üldine skeem MODIS päevase GPP ja aastase NPP arvutamiseks on ära toodud joonisel 3. Päevast NPP'd ei arvutata kuna kõiki

GPP'st NPP saamiseks vajaminevaid päevase hingamise komponente ei ole võimalik päevase ajasammuga arvutada. Küll aga on need hinnatavad terve aasta jaoks.



Joonis 3. MODIS päevase GPP ja aastase NPP algoritmi loogika [5].

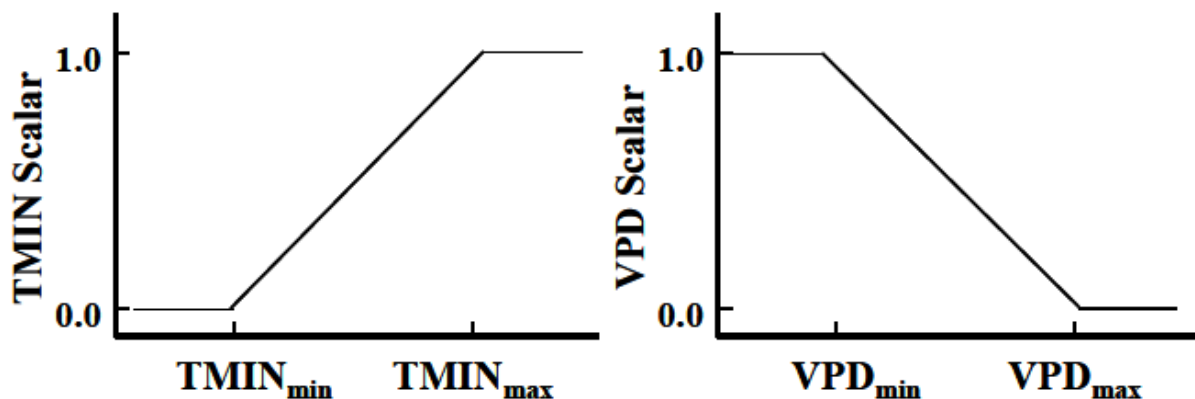
Alustatakse päevase GPP arvutamisega, mis koosneb kahest peamisest arvutuskäigust, millede korrutamisel saadaksegi päevane GPP.

Esimeses leitakse iga mudeli võrgusilma jaoks vastava päeva kiirguse kasutamise efektiivsus. Selleks on eelnevalt iga maakattetüübi jaoks leitud 5 parameetrit, mida on kirjeldatud tabelis 1.

Tabel 1. Päevase KKE arvutamiseks vajaminevad parameetrid.

Parameeter	Ühik	Kirjeldus
ϵ_{\max}	$\text{kgC} \cdot \text{MJ}^{-1}$	Maksimaalne kiirguse kasutamise efektiivsus
T_{\min}^{start}	$^{\circ}\text{C}$	Ööpäevane minimaalne temperatuur, mille puhul $\epsilon = \epsilon_{\max}$
T_{\min}^{full}	$^{\circ}\text{C}$	Ööpäevane minimaalne temperatuur, mille puhul $\epsilon = 0$
VPD^{start}	Pa	Päevane keskmine veeaururõhu defitsiit, mille puhul $\epsilon = \epsilon_{\max}$
VPD^{full}	Pa	Päevane keskmine veeaururõhu defitsiit, mille puhul $\epsilon = 0$

Nendest 4 viimast parameetrit määravad maksimaalset KKE-d vähendavad ilmastikust sõltuvad piirangufunktsioonid, mille üldised kujud on ära toodud joonisel 4.



Joonis 4. MODIS GPP algoritmi maksimaalset kiirguse kasutamise efektiivsuse väärtust vähendavate piirangufunktsioonide üldised kujud, mille rakendamisel leitakse vastava päeva efektiivne kiirguse kasutamise efektiivsus [5].

Tabelis 1 ära toodud parameetrite väärtused leiti läbi optimeerimisprotsessi, mis minimeeris protsessipõhise Biome-BGC mudeli [21] väljundite ja MODIS GPP algoritmi tulemuste absoluutset vahet. Biome-BGC on vee, süsiniku ja lämmastiku voogusid süsteemis taime- ja põhk-muld-maapinnalähedane atmosfäär kirjeldav protsessipõhine mudel.

Seega leitakse päevane KKE MODIS algoritmis valemiga

$$\epsilon = \epsilon_{\max} \times \text{TMIN_Scalar} \times \text{VPD_Scalar}, \quad (3)$$

mis lisaks tabelis 1 toodud parameetritele vajab sisendina ka meteoroloogilisi andmeid, kust arvutada vajaminevad ööpäeva minimaalne temperatuur (TMIN) ja päevane keskmine veeaururõhu defitsiit (VPD). Need saadakse numbrilise ilmamudeli väljundist.

Päevane taime- ja põhk-muld-maapinnalähedane atmosfäär kirjeldav protsessipõhine mudel poolt neelatud fotosünteesiliselt aktiivse kiirguse hulk (APAR ehk inglise keeles *Absorbed Photosynthetically Active Radiation*) leitakse valemiga

$$\text{APAR} = \text{fAPAR} \times \text{PAR} = \text{fAPAR} \times \text{SWRad} \times 0.45, \text{ kus} \quad (4)$$

SWRad on lühilainelise päikesekiirguse hulk (MJ), mis on samuti ilmamudeli väljundite hulgas. fAPAR saadakse MODIS fAPAR produktist.

Nii on päevane GPP leitav kui

$$\text{GPP} = \text{APAR} \times \epsilon = \text{fAPAR} \times \text{SWRad} \times 0.45 \times \epsilon_{\max} \times \text{TMIN_Scalar} \times \text{VPD_Scalar} \quad (5)$$

ja aastane GPP saadakse lihtsalt päevaste GPP väärtuste summeerimisel üle aasta.

NPP arvutamiseks GPP-st on vaja maha lahutada rakuhingamise ja kasvu hingamise tõttu atmosfääri paisatud süsiniku hulk. Päevaselt ja operatiivselt on MODIS algoritmi ülesehituses arvutatavad lehtede ja juurte rakuhingamine. Puidu rakuhingamine ja taimeistiku kasvu hingamise arvutamiseks ei ole olemas vajalikke sisendandmeid või valemideid, küll aga on need hinnatavad tagasiulatavalt omades terve ühe vegetatsiooniperioodi sisendandmeid - puidu rakuhingamise puhul iga päeva kohta, aga kasvu hingamise puhul vaid hinnanguna terve aasta summa jaoks.

Hingamise komponentide arvutamiseks MODIS NPP algoritmis on olemas erinevad rakuhingamist kirjeldavad valemid ja maakattetüübist sõltuvad valemitesse minevad parameetrid. Toodetavatest sisenditest on tarvis päevast keskmist temperatuuri ja lehepinnaindeksi (LAI ehk inglise keeles *Leaf Area Index*) väärtust iga mudeli võrgusilma jaoks. LAI võetakse MODIS'e LAI produktist.

Raku hingamise arvutamine vastavalt maakattetüübile puidu, lehtede ja juurte jaoks käib järgmise üldise valemi järgi

$$\mathbf{MR}_X = \mathbf{m}_X \times \mathbf{MR}_{X_base} \times \mathbf{Q}_{10_X_MR}^{\frac{T_{avg}-20}{10}}, \text{ kus} \quad (6)$$

kus indeks X tähistab, kas tegemist on puidu, lehtede või juurtega, \mathbf{MR}_X = raku hingamine ($\text{kgC} \cdot \text{päev}^{-1}$), \mathbf{m}_X = mass (kg), \mathbf{MR}_{X_base} = raku hingamine massiühiku kohta päevas ($\text{kgC} \cdot \text{kgC}^{-1} \cdot \text{päev}^{-1}$), \mathbf{Q}_{10_MR} = eksponentfunktsiooni kuju kontrolliv parameeter ja T_{avg} = ööpäeva keskmine temperatuur ($^{\circ}\text{C}$). Lehtede jaoks leitakse $\mathbf{Q}_{10_X_MR}$ valemist

$$\mathbf{Q}_{10_X_MR} = \mathbf{Q}_{10_Leaf_MR} = 3.22 - 0.046 \times T_{avg}, \quad (7)$$

puidu ja juurte jaoks on tegemist konstandiga. Lehtede ja juurte mass arvutatakse LAI väärtusest vastavate allomeetriliste valemite järgi. Puidu raku hingamist, vastupidiselt lehtede ja juurte omale, ei saa operatiivselt arvutada kuna puidu mass ei arvutata LAI hetkeväärtuste põhjal vaid LAI aastase maksimaalse väärtuse põhjal. Küll on aga puidu päevased raku hingamised järelarvutatavad.

Kasvu hingamise arvesse võtmine käib MODIS algoritmis lihtsa järgneva valemi järgi

$$\mathbf{GR} = 0.25 \times \mathbf{NPP}, \text{ kus} \quad (8)$$

GR = kasvu hingamine ($\text{kgC} \cdot \text{päev}^{-1}$). Nii on aastane NPP arvutatav kui

$$\mathbf{NPP} = \mathbf{GPP} - \mathbf{MR} - \mathbf{GR} = \mathbf{NPP} - \mathbf{MR} - 0.25 \times \mathbf{NPP}, \text{ kus} \quad (9)$$

NPP = aastane netoproduktsoon ($\text{kgC} \cdot \text{aasta}^{-1}$), GPP = aastane brutoproduktsoon ($\text{kgC} \cdot \text{aasta}^{-1}$), MR = aastane rakuhingamine ($\text{kgC} \cdot \text{aasta}^{-1}$) ja GR = aastane kasvuhingamine ($\text{kgC} \cdot \text{aasta}^{-1}$). Kuna aastane GPP võib ka negatiivne olla, on lõplik aastase NPP arvutuskäik

$$\text{NPP} = 0.8 \times (\text{GPP} - \text{MR}) \quad , \text{ kui } \text{GPP} - \text{MR} \geq 0 \text{ ja} \quad (10)$$

$$\text{NPP} = 0 \quad , \text{ kui } \text{GPP} - \text{MR} < 0. \quad (11)$$

2.3. EST-PP

EST-PP on MODIS GPP ja NPP algoritmide põhjal loodud primaarproduktiooni mudel Eesti ala jaoks [13]. Algoritmide poole pealt on ta MODIS'ele väga sarnane, peamine erinevus on sisendandmetes. Järgnevalt keskendumegi ainult nende kahe mudeli algoritmide ja produktide erinevuste kirjeldamisele.

Suurima mõjuga erinevus võrrandites kahe mudeli vahel on temperatuuri piirangufunktsiooni $g(T)$ definitsioon, mis EST-PP puhul on kirjeldatav järgnevate valemitega

$$g(T) = \frac{(T - T_{\min})(T - T_{\max})}{(T - T_{\min})(T - T_{\max}) - (T - T_{\text{opt}})^2}, \text{ kui } T_{\min} < T < T_{\max} \quad (12)$$

$$g(T) = 0, \text{ kui } T \leq T_{\min} \text{ või } T \geq T_{\max}, \text{ kus} \quad (13)$$

$g(T)$ = temperatuuri piirangufunktsioon, T = ööpäeva keskmine temperatuur ($^{\circ}\text{C}$), T_{\min} = ööpäeva keskmine temperatuur ($^{\circ}\text{C}$), millest alates temperatuuri piirangufunktsioon hakkab kasvama, T_{opt} = ööpäeva keskmine temperatuur, mille puhul temperatuuri piirangufunktsioon saavutab oma maksimaalse väärtuse ($^{\circ}\text{C}$) ja T_{\max} = ööpäeva keskmine temperatuur, mille juures temperatuuri piirangufunktsioon lõpetab kahanemise ja saab jälle väärtuseks 0 ($^{\circ}\text{C}$). Selle üldist kuju kirjeldab joonis 5. Lisaks piirangufunktsiooni kujule seisneb erinevus MODIS'est seega ka kasutatavas temperatuuris – ööpäeva keskmine EST-PP puhul ja ööpäeva minimaalne MODIS'e puhul. Selline piirangufunktsioon on taimefüsioloogilisest vaatevinklist rohkem põhjendatud, kui MODIS'e oma, sest negatiivsete temperatuuri väärtuste juures taimede elutegevust ei toimu.

Teine erinevus võrrandites on vähem seotud teoreetiliste ja taimefüsioloogiliste kaalutlustega ning rohkem seotud sisendandmetike valikust tulevate omapäradega. Erinevus seisneb taimkattele langeva Päikese lühilainelise kiirgushulga arvutamismetoodikas. Kui MODIS'e implementatsioonis võetaks see suurus ilnamudeli väljundist, siis EST-PP puhul kasutatakse Tõraveres mõõdetud kiirgushulkasid ja pilvisust, mida kasutatakse koos ilnamudelist saadud pilvisuse andmetega, et hinnata kiirgushulka terve Eesti ehk modelleerimise ala peal.

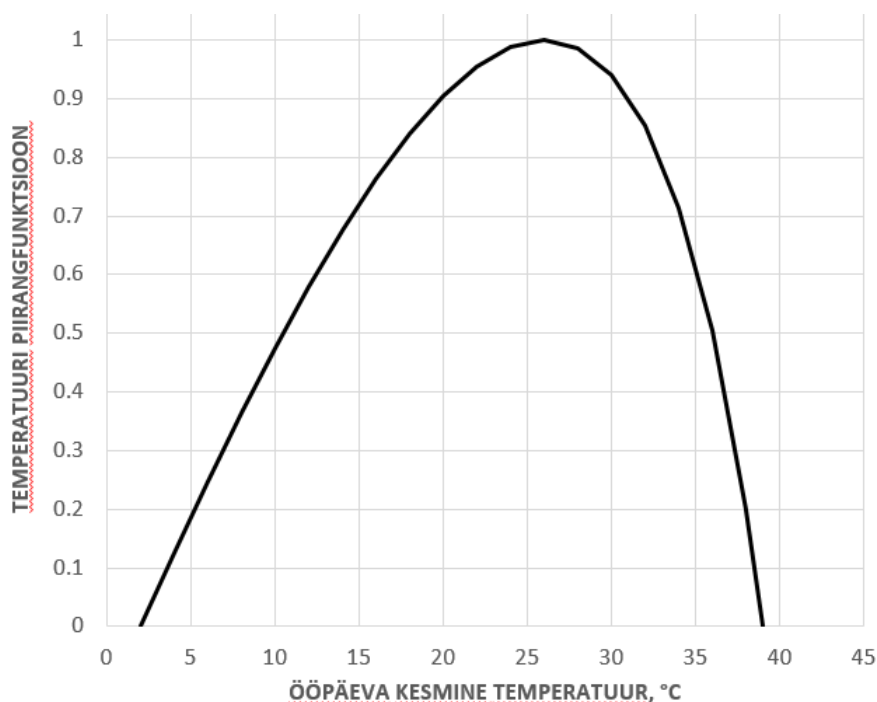
Päevase summaarse lühilainelise Päikesekiirguse arvutamine iga mudeli võrgusilma jaoks käib valemiga

$$\text{SWRad}(t) = Q_{\text{measured}, \text{Tõravere}}(t) \frac{\text{CCF}(\text{CC}(t))}{\text{CCF}(\text{CC}_{\text{Tõravere}}(t))}, \text{ kus} \quad (14)$$

SWRad = summaarne päevane lühilainelise Päikesekiirguse hulk (MJ), t = päev aastas, $Q_{\text{measured}, \text{Tõravere}}$ = Tõravere mõõdetud summaarne päevane lühilainelise päikesekiirguse hulk (MJ), CC = ilmapudelit väljundist saadud pilvisuse koefitsient, $\text{CC}_{\text{Tõravere}}$ = Tõravere mõõtmistest saadud pilvisuse faktor ja CCF = pilvisuse koefitsient, mille arvutamiseks kasutatakse valemit

$$\text{CCF} = -0.006788 \times \text{CC}^2 + 0.010986 \times \text{CC} + 0.953658. \quad (15)$$

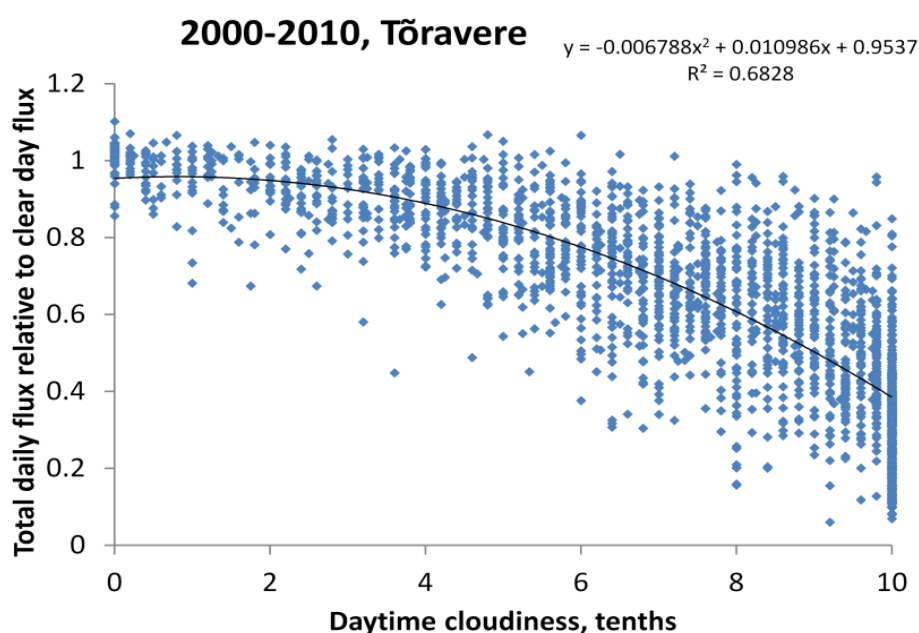
Pilvisuse koefitsiendi arvutusvalem leiti Tõravere läbi viidud päevase summaarse Päikesekiirguse ja pilvisuse mõõtmistest aasta 100. päevast kuni 290. päevani ehk siis üldiselt lumevabast perioodist. Seda andmestikku kirjeldab joonis 6 (autor Tiit Nilson).



Joonis 5. EST-PP temperatuuri piirangufunktsiooni kuju.

Rohkem algoritmilisi erinevusi nende kahe mudeli vahel pole. Küll on aga erinevad nende sisendandmete allikad. Kui MODIS kasutab fAPAR ja LAI väärtuste aegridade allikatena MODIS vastavaidprodukte, siis EST-PP kasutab MERIS instrumendi [22] andmetest loodud vastavaidprodukte. Samuti erinevad meteoroloogiliste andmete allikad. Lisaks juba mainitud pilvisuse ja päevase summaarse kiirgushulga kohtmõõtmistele, on erinevused ka kasutatud

ilmamudelites. Ehk siis nii mudelite lahendusblokkides kui ka väljundite ruumilises lahutuses. Nimelt on EST-PP sisendina kasutatud meteoroloogilised andmed osalt BaltAn65+ reanalüüsist [23], osalt endise EMHI, nüüdse Keskkonnaagentuuri, HIRLAM mudelist [24]. Mõlemad on loodud Läänemereäärse regionaalse ilma modelleerimiseks, seega kasu võiks tuleneda nii parematest regionaalsetest algoritmidest kui ruumilisest lahutusest. MODIS saab aga oma meteoroloogilised sisendid NASA GMAO'st (*Global Modeling and Assimilation Office*) [25]. Suurim kasu tuleb just ruumilisest lahutusest, mis võimaldab Eesti ala piires regionaalseid ilmastiku erinevusi palju täpsemalt modelleerida.



Joonis 6. Pilvisuse koefitsiendi arvutamine.

Tähelepanuväärne erinevus on ka sisendina kasutatud maakattekaardil. MODIS kasutab oma maakatte produkti [26], mis luuakse automaatse töötamise käigus [27] ja mille kvaliteet seetõttu kannatab, ning mille ruumiline lahutus on 1 km². EST-PP jaoks lõi autor poolautomaatselt algselt Eesti detailne maakatte kaart 22 m lahutusega satelliidipiltidest ja erinevatest ruumiandmetest (hooned, teed, PRIA põllumaad ja rohumaad), mis siis mudeli 1 km² võrgustikule sai skaleeritud.

2.4. Peamised erinevused mudelite vahel

Viimastel aastatel on teaduskirjandusse ilmunud mitmeid erinevaid KKE tüüpi GPP ja NPP mudeleid [7] [9] [10] [1] [12]. Üks aspekt, milles need erinevad, on kasutatavad sisendandmete allikad. See ei ole siiski märkimisväärne külg, sest kõik need mudelid nõuavad eelnevalt mõnede parameetrite leidmist referentsandmestike põhjal ja seega väheneb kasutatud

sisendandmete mõju tulemustele. Peamiselt on mudelite käitumise ja tulemuste erinevuste põhjuseks kasutatud piirangufunktsioonid – nii funktsioonide kujud kui ka nendes kasutatavad meteoroloogilised või muud keskkonda ja kasvutingimusi iseloomustavad suurused. Kuna mitmed mudelid on üles ehitatud suurustele, mis on suhteliselt standardised ilnamudelite väljundid, on viimaste aastate jooksul olnud võimalik neid omavahel võrrelda [1] [2] [11] [12] [3]. Selleks kasutatakse kõigi puhul samasid meteoroloogilisi sisendandmeid ja GPP ning NPP referentsandmestike, mida kasutatakse iga mudeli jaoks parimaid tulemusi andva parameetrite komplekti leidmiseks. Seejärel jooksutatakse uuritavad mudelid sama ala peal ja võrreldakse nende tulemusi, nii huvipakkuvate suuruste summaarseid kui ka ajalisi ja ruumilisi erinevusi.

Nii on võimalik hinnata erinevate implementatsioonide sobivust erinevatesse ökosüsteemidesse. Omajagu uuritakse siiski ka kasutatavate sisendandmete sobilikkust, eelkõige just erinevate satelliitproduktide mõju fAPAR väärtustele ja ajalisele käigule.

Piirangufunktsioonid võib üldiselt kahte klassi jagada: need, mis arvestavad temperatuuri mõju ja need, mis arvestavad veerežiimi mõju primaarproduksioonile.

Temperatuuri piirangufunktsioonid vajavad sisendina üldjuhul ööpäeva või päeva keskmist või minimaalset temperatuuri, kas 2 m kõrgusel atmosfääris või maapinnal. Piirangufunktsioonide kujud on enamjaolt juba varem kirjeldatud MODIS või EST-PP mudelites kasutatute sarnased. On ka keerukamaid lahendusi, näiteks on subarktilise ja parasvöötme jaoks häid tulemusi andnud puude temperatuuri aklimatsiooni arvesse võtmine [9].

Suuremad erinevused mudelite vahel on veerežiimist tuleneva piirangu implementatsioonides. Esiteks eksisteerib rohkem erinevaid suurusi, mida selle hindamiseks saab kasutada. Mitme mudeli efektiivse KKE arvutamise valemid ja veerežiimi piirangufunktsioonid on ära toodud tabelis 2 [8].

Nagu näha, on veerežiimi piirangufunktsiooni argumendiks kasutatud veeaururõhu defitsiiti (tabelis tähistatud D-ga), maapinna vee indeksit (LSWI ehk inglise keeles *Land Surface Water Index*), pinnase niiskussisaldust (SM ehk inglise keeles *Soil Moisture*), evapotranspiratsiooni (E), erinevaid potentsiaalse aurumise definitsioone (E_{Th} ja E_{PT}) ning kiirgushulka (R_n). Kuna piirangufunktsiooni väärtus peab olema vahemikus [0,1], siis tihtipeale kasutatakse mingi suuruse reaalse ja potentsiaalse väärtuse suhet. TOPS, 3-PG ja CFLUX mudelid leiavad kahe erineva piirangufunktsiooni väärtused ja rakendavad efektiivse KKE arvutamiseks neist madalamat.

Tabel 2. Efektiivse KKE ja veerežiimi piirangufktsiooni arvutamine erinevates KKE tüüpi primaarproduktiooni mudelites [8].

Model	ε (g CMJ ⁻¹) ^a	W_ε
TURC	$\varepsilon_g = \varepsilon^*$	No
C-Fix	$\varepsilon_g = \varepsilon^* \times T_\varepsilon$	No
MOD17	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon$	$W_\varepsilon = \frac{D_{\max} - D}{D_{\max} - D_{\min}}$
VPM	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon$	$W_\varepsilon = \frac{(1 + \text{LSWI})}{(1 + \text{LSWI}_{\max})}$
BEAMS	$\varepsilon_g = \varepsilon^* \times P_{\text{actual}} \times P_{\max}$	$W_\varepsilon = P_{\text{actual}} / P_{\max}$
GLO-PEM	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon$	$W_\varepsilon = \min(f(\text{SM}), f(D))$
TOPS	$\varepsilon_g = \varepsilon^* \times \min(T_\varepsilon, W_\varepsilon)$	
3-PG	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon \times S_a$	$W_\varepsilon = \min(f(\text{SM}), f(D))$
CFLUX	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon \times S_a$	$W_\varepsilon = \min(f(\text{SM}), f(D))$
CASA	$\varepsilon_n = \varepsilon^* \times T_{\varepsilon 1} \times T_{\varepsilon 2} \times W_\varepsilon$	$W_\varepsilon = 0.5 + 0.5E/E_{\text{Th}}$
EC-LUE	$\varepsilon_g = \varepsilon^* \times \min(T_\varepsilon, W_\varepsilon)$	$W_\varepsilon = E/R_n$
TEC	$\varepsilon_g = \varepsilon^* \times T_\varepsilon \times W_\varepsilon$	$W_\varepsilon = E/E_{\text{PT}}$

3. EST-PP-Python

3.1. Taust

EST-PP-Python on EST-PP mudeli rakendus programmeerimiskeeles Python. Eelnevalt oli EST-PP mudeli tulemused saadud programmeerimiskeeles IDL (*Interactive Data Language*) kirjutatud individuaalsete skriptide rakendamisel sisendandmetele või vahetulemustele [13]. Kuna iga skript defineeris ise erinevad vajalikud sisendparameetrid ja terve töötlusprotsessi läbi tegemiseks oli vaja igaühe individuaalset manuaalset käivitamist, ei olnud tegemist praktilise lahendusega, mida pikemaks teadustööks või õppetöös kasutada. Seetõttu sai otsustatud luua täiesti uus rakendus.

3.2. Arenduse põhimõtted

EST-PP-Python peamine eesmärk oli luua tarkvara, mida oleks mugav ja praktiline kasutada primaarproduktiooni uurimiseks või vastavas õppetöös. Selleks lähtuti järgnevatest põhimõtetest:

- 1) Töötlusprotsess peab olema algusest lõpuni automaatne
- 2) Tarkvara paigaldamine peab olema võimalikult lihtne
- 3) Tarkvara käivitamine peab olema piisavalt lihtne, et sellega saaksid hakkama ka väheste arvutikasutamise oskustega inimesed
- 4) Tarkvara/mudeli konfigureerimine peab olema võimalikult lihtne, et sellega saaksid hakkama ka väheste arvutikasutamise oskustega inimesed
- 5) Samas peaks olema võimalik sellise konfigureerimise teel muuta osaliselt mudeli käitumist, näiteks:
 - Muuta sisendandmestikku(e)
 - Muuta piirangufunktsioone
 - Muuta soovitud väljundeid
 - Muuta modelleerimisala
 - Muuta modelleerimise ajavahemikku
- 6) Tarkvara peab kasutama vabavaralisi lahendusi
- 7) Mudeli jooksutamise protsessid ja ülesehitus peaksid olema niimoodi kirjutatud, et mudel oleks kergesti laiendatav, näiteks uute piirangufunktsiooni lisamisega, ilma, et vajaks suurt ümber kirjutamist vaid pigem lokaalseid isoleeritud täiendusi

- 8) Tarkvara peab olema Python'is imporditav Python paketina, mille klassid ja meetodid oleksid kasutatavad kasutaja enda GPP või NPP arvutamise töötlusprotsesside loomiseks
- 9) Eelkõige peab ta töötama samade sisendandmete peal, kui EST-PP. Samas peab ta olema kirjutatud nii, et selle käivitamine teiste sisendandmete pealt ei nõuaks üldse või üleliia lisatööd

3.3. Arenduse keskkond

Esimese asjana tuli otsustada kasutatav programmeerimiskeel. Arenduse eesmärges silmas pidades sai otsustatud Python'i kasuks kuna tegemist on vabavaralise programmeerimiskeelega, teaduses ja üliõpilaste seas laialt levinud oma mitmete teadusele suunatud pakettide olemasolu poolest ja tegemist on suhteliselt lihtsa keelega, mis sobib väheste programmeerimiskogemustega inimestele töötamiseks ja arenemiseks. Järgnevalt sai valitud, kas kasutada Python 2-te, täpsemalt 2.7-t, või Python 3-e, täpsemalt 3.5-te (sellel ajal kõige uuem versioon). Python 2.7-e kasuks rääkis tema küpsus ja suurem kasutatavus - mitmed kasutajate poolt loodud Python'i (teadus)paketid ei toetanud/toeta Python 3-e kuna nende arendusega hakati pihta enne kolme publitseerimist. Python 3-e puhul oli argumendiks, et tegemist on aktiivselt arendatava versiooniga ja tegemist on Python'i „tulevikuga“ ning peamiselt seetõttu sai see ka valituks. Arenduse tarkvarana kasutati ettevõtte JetBrains vabavaralist integreeritud arenduskeskkonda PyCharm Community Edition [28]. Koodihoidlana kasutati BitBucket'it [29], versioonikontrollisüsteemina Git'i [30] ja selle töölaua klienti SourceTree [31] ettevõttelt Atlassian [32]. Kõik need on täielikult või osadel tingimustel vabavaralised lahendused.

3.4. Arenduse ülesehitus

EST-PP-Python struktuur põhineb Python programmide ülesehituse heal taval. Kõrgeim tase sisaldab kaustu *estpp* ja *scripts*, standardseid tarkvara kirjeldavaid faile (*.gitignore*, *README.md*, *contributors.txt* ja *requirements.txt*) ja installeerimise skripti. Ülesehituse loomisel sai silmas peetud, et soovi korral oleks võimalik EST-PP-Python PyPI-sse (Python Package Index) üles laadida, kust see oleks alla laetav ja installeeritav terve Python'i kogukonna poolt. EST-PP-Python Python'i paketi (nimega *estpp*) struktuuri kirjeldab ka joonis 7.

Kaust *estpp* sisaldab endas mudeli põhikoodi. Seal asuvad moodulid *config.py*, *estpp.py*, *gpp.py*, *npp.py*, *outputs_processing.py*, ja *utils.py*. Nendest tuleb lähemalt juttu hiljem tarkvara funktsionaalsusi käsitlevas peatükis. Lisaks veel inimloetavas failiformaadis vaikimisi väärtustega konfiguratsioonifail *default_config.yaml*, näiteks toodud täidetud

konfiguratsioonifail *user_config.py* ning kaks Python keelele omast faili: *__init__.py* ja *__main__.py*.

Kaustas *scripts* asub mudeli käsurealt käivitamisel välja kutsutav skript *run_estpp.py* ja mudeli väljundite analüüsimiseks ja graafikute loomiseks kasutatav näidisskript *outputs_analysis.py*.

```
estpp/  
  estpp/  
    __init__.py  
    __main__.py  
    config.py  
    default_config.yaml  
    estpp.py  
    gpp.py  
    npp.py  
    outputs_processing.py  
    user_config.yaml  
    utils.py  
  scripts/  
    outputs_analysis.py  
    run_estpp.py  
  .gitignore  
  README.md  
  contributors.txt  
  requirements.txt  
  setup.py
```

Joonis 7. EST-PP-Python Python paketi, nimega *estpp*, struktuur.

Mudeli põhikood koosneb umbkaudu 2000-st koodireast, millest enamus on enamvähem võrdselt ära jaotatud moodulite *config.py*, *gpp.py*, *npp.py* ja *outputs_processing.py* vahel.

3.5. Kasutamine

Tavakasutaja jaoks koosneb mudeli käimapanek kahest sammust: üldise konfiguratsioonifaili täitmisest ja käsurealt mudeli käima panemisest andes argumentina ette eelnevalt täidetud konfiguratsioonifaili.

Konfiguratsioonifail on inimloetavas YAML formaadis ja seal määratavad suurused on ära toodud lisas 1. Eelkõige määratakse seal ära sisendandmete asukohad, modelleerimise ala ja ajavahemik ning soovitud väljundid koos väljundkausta asukohaga. Ka piirangufunktsioonide tüübid saab seal määrata.

Sisenditest vajab määramist maakattekaardi, Tõravere kiirgusmõõtmiste ja maakatteklassidest sõltuvate parameetrite tabeli asukohad. Samuti lehepinnaindeksi, fAPARi, temperatuuri, suhtelise õhuniiskuse ja pilvisuse andmeid sisaldavad kaustade asukohad.

Modelleerimise ala määramiseks on kaks võimalust. Ühena võib ette anda teegi GDAL (Geospatial Data Abstraction Library) poolt loetava rasterandmestiku, mille asukoha, mõõtmete ja piksli suuruste järgi määratakse siis vastav modelleerimise ala. Teise võimalusena võib need ise manuaalselt sisestada konfiguratsioonifaili määrates projektsiooni, ridade ning tulpade arvu ja vastavad *geotransform* (vt lähemalt GDAL dokumentatsioonist) väärtused.

Modelleerimise ajavahemiku määramine käib järgneva malli alusel: YYYY-MM-DD – YYYY-MM-DD, määratud kuupäevad kaasaarvatud.

Väljundite määramisel tuleb eelkõige täita andmed väljundkausta asukohaga ja soovi korral ka vastavat mudeli jooksu iseloomustava jooksu nimega. Lisaks on valida 22 erineva väljundi vahel. Osad neist iseloomustavad vastava jooksu konfiguratsiooni jms (näiteks logi fail), enamused aga on erinevad mudeli vahe- või lõpptulemid (GPP, NPP, keskmine ööpäeva temperatuur, lehtede rakuhingamine, fAPAR jne.). Enamus neist väljastatakse päevase ajasammuga vastavalt mudeli ajalisele lahutusele. Kuigi näiteks fAPAR väärtused on olemas ka sisendandmete all, siis nende kirjutamine väljundite kausta tuleneb mõttest, et iga jooksu väljundkaust oleks täiesti eraldiseisev andmestik, mille pealt järgnevaid analüüse teha. See lihtsustab tulemuste jagamist, andmete organiseerimist ja ka tulemuste analüüsi, sest kõik väljundid on analoogsed failiformaadis. Samuti aitab see valideerida mudeli korrektset käitumist.

Kui mõnda suurust kasutaja konfiguratsioonifailis määratud pole, siis võetakse selle vaikimisi väärtus (nendel parameetritel, millel see eksisteerib) failist *default_config.yaml*. Kui seda ka sealt ei leita, väljastatakse veateade ja palutakse konfiguratsioonifaili korrektselt täita.

Loomulikult on EST-PP-Python Python'i pakett *estpp* ja selle erinevad moodulid (*gpp.py*, *npp.py*) koos neis sisalduvate klasside ja meetoditega ka imporditavad enda arendusse, mis võimaldab nende põhjal ka enda töötlusprotsessi üles ehitada. Sellist kasutusjuhtu sai arenduse käigus ka silmas peetud, mis väljendub klasside ja nende meetodite valikus. Küll aga peab meele pidama, et praegune arendus on eelkõige kirjutatud EST-PP mudeli ja selle sisendandmete peale ning mõningasest lisaarendusest, kui on soovi mudelit enda juhu järgi kohendada, ilmselt pääsu pole.

3.6. Tööpõhimõte ja funktsionaalsus

Järgnevalt saavad lähemalt kirjeldatud *estpp* erinevad moodulid – nende eesmärk ja funktsionaalsus läbi klasside ja meetodite. Järjekord, milles neid tutvustatakse, põhineb nende sisu välja kutsumise järjekorral pärast kasutaja poolt mudeli käivitamist käsurealt, eeldades, et väljunditena on nõutud nii brutoproduktiooni kui ka netoproduktiooni hinnangud. Selline lähenemine mitte ei võimalda tutvustada iga mooduli funktsionaalsust, vaid seob nad kokku ka üheks terviklikus töötusprotsessiks. Moodulid, nende kirjeldamise järjekorras, on: *estpp.py*, *config.py*, *gpp.py*, *npp.py* ja *outputs_processing.py*. Samuti seletatakse eraldi skripti *outputs_analysis.py* mõtet.

3.6.1. *estpp.py*

Selles leiduv funktsioon *main()* kutsutaksegi käsurealt käivitatava *run_estpp.py* skripti poolt välja. Lisaks sellele funktsioonile defineeritakse siin moodulis ka klass *ESTPP* koos tema ainukese meetodi *run()*-ga.

main() funktsioon teeb kolme asja (vaata ka joonis 8):

- 1) Loeb käsurealt antud kasutaja konfiguratsioonifaili asukoha.
- 2) Loob selle põhjal töötuseks vajamineva jooksu sisemise konfiguratsiooni. Selleks kasutab ta mooduli *config.py* funktsionaalsust, millest täpsemalt hiljem.
- 3) Algväärtustab klassi *ESTPP* ühe isendi (isend on mingi klassi üks instant) andes talle argumendiks eelnevalt loodud detailse jooksuprojektsiooni ja kutsub välja selle meetodi *run()*.

Klassimeetod *ESTPP().run()* on iseenesest lühike ja lihtne ning selle peamine funktsionaalsus seisneb moodulite *gpp.py* ja *npp.py* *run()* funktsioonide välja kutsumises vastavalt konfiguratsioonifailis märgitud soovitud väljunditele.

```
def main():
    arg_parser = argparse.ArgumentParser(description='EST-PP-Python primaarproduktiooni mudel')

    arg_parser.add_argument(
        '-v', '--version',
        action='version',
        version=__version__    '#{0} {1}'.format(metadata.project, metadata.version))
    arg_parser.add_argument("config_file",
                            help="The configuration Yaml file.")
    args = arg_parser.parse_args()

    run_config = Config()
    run_config.update_with_user_config(args.config_file)
    run_config.initialise()

    ESTPP(run_config).run()
```

Joonis 8. *main()* funktsioon ning *Config* klassi isendi loomine ja peamine kasutamisiis.

3.6.2. config.py

Config.py puhul on tegemist tarkvara tuumaga, sest temas leiduva ainukese klassi, nimega *Config*, isend on vajalik sisend kõikide teiste objektide loomiseks. *Config* klassi muutujad ja meetodid on välja toodud joonisel 9.

```
def __init__(self):
    self.default_general_config = Config.default_general_config.copy()
    self.general_config = Config.default_general_config.copy()
    self.start_date = date
    self.end_date = date
    self.modelling_area = pd.Series()
    self.veg_indices_paths = pd.DataFrame()
    self.meteo_paths = pd.DataFrame()
    self.measured_PAR = pd.DataFrame()
    self.BPLUT = pd.DataFrame()
    self.land_cover_path = Path()
    self.valid_days = pd.DataFrame()
    self.outputs = pd.DataFrame()
    self.outputs_file_format = 'GTiff'
    self.daily_outputs_paths = pd.DataFrame()
    self.annual_outputs_paths = pd.DataFrame()
    self.runs_dir = Path()
    self.run_name = str()
    self.run_dir = Path()
    self.out_nodata_value = -9999
    self.temp_stress_factor_function = str()
    self.water_stress_factor_function = str()

    @classmethod
    def dump_default_config(cls):

    def load(self, config_file)

    def dump(self, config_file)

    def update_with_user_config(self, config_file)

    @staticmethod
    def update_nested_dict(target, updates)

    def initialise(self)

    def check_configuration_validity(self)

    def create_detailed_config(self)

    @staticmethod
    def datespan(start_date, end_date, delta=timedelta(days=1))
```

Joonis 9. *Config* klassi muutjad ja meetodid. Tähtsaimad meetodid on punase piirjoonega esile tõstetud.

Config klassi kasutamist näitlikustavad hästi mooduli *estpp.py main()* funktsiooni viimased read (joonis 8). Alguses luuakse üks *Config* klassi isend. Seejärel uuendatakse temas üldist jookskonfiguratsiooni kirjeldava sõnaraamatu tüüpi muutuja andmed vastavalt kasutaja poolt täidetud ja ette antud konfiguratsioonile. Lõpuks kutsutakse välja meetod *initialise()*, mis kujutab endast kahe teise meetodi, *check_configuration_validity()* ja *create_detailed_config()*, järjestikust välja kutsumist. Neist esimene viib läbi mõned kontrollid veendumaks, et kasutaja poolt ette antud konfiguratsioon on korrektne ja teine loob selle konfiguratsiooni põhjal sisemiseks kasutuseks detailse jookskonfiguratsiooni, mis seisneb isendi algväärtustamisel loodud muutujatele sisu loomises. Kõik muutujad koos nende tüübiga on nähtavad joonise 9 vasakus tulbas.

Meetodi *create_detailed_config()* üks tähtsamaid funktsioone on, et ta käib läbi üldises konfiguratsioonis sätestatud sisendandmete kaustad ja kontrollib sisendandmete olemasolu ning salvestab iga faili asukoha andmebaasi hilisemaks kiireks pärimiseks. Samuti loob ta valiidsete päevade andmestiku, mis kujutab endast infot päevadest, mille kohta on piisavalt sisendandmeid olemas, et tulemused arvutada. See on väga vajalik pikkade ajaperioodide ja suure hulga andmete puhul, sest tihti peale esineb andmeridades lünkasid. Nii saab kohe mudeli jooksumise alguses kontrollitud, et kõik mudeli jooksumiseks vajalikud andmed on olemas

või mis ulatuses nad eksisteerivad, ja vajadusel jooksu katkestada. Valiidsete päevade andmestikku kasutatakse hiljem mudeli üle päevade jooksumisel, et kohe jätta vahele päevad, mille kohta tulemusi genereerida pole võimalik. See võib endas märgatavat ajakulu võitu tähendada.

Create_detailed_config() on ka peamine meetod, mida muuta, kui soovida mudelit jooksutada mingite muude sisendandmete pealt. Praeguses rakenduses oskab vastav meetod lugeda kindla failinime formaadiga kõiki GDAL teegi poolt avatavaid rasterandmeid. Ehk siis, kui soov on kasutada samu füüsikalisi suuruseid, aga teisi andmeid, piisab ka andmete õigesse formaati töötlemisest ja mudel peaks olema käivituv. Kui aga tahta lisada mõnda täiesti uut andmekomplekti, siis peab lisama vastava isendimuutuja klassi *Config __init__()* meetodisse ja kirjutama meetodisse *create_detailed_config()* juurde nende sisse lugemise ja eelnevalt lisatud muutuja väärtustamise. Enamus *Config* klassi isendi muutujatest on *Pandas Dataframe* tüüpi, sest nendest on lihtsa ja loetava süntaksiga kerge konkreetseid väärtusi pärida.

3.6.3. gpp.py

Brutoproduktiooni arvutamise moodul, mis sisaldab endas klasse *GPP*, *TemperatureStressFactor* ja *WaterStressFactor*.

Klassi *GPP* meetodeid ja algväärtustamist kirjeldab joonis 10. Enamus meetodeid tegelevad kas *GPP* leidmiseks vajaminevate suuruste arvutamisega või väljundite kõvakettale kirjutamisega. Peamiseks sisendiks enamike meetodite jaoks on kuupäev, mille jaoks vastavad suurused arvutada. Selle järgi pärivad meetodid jookskonfiguratsioonist enda arvutusteks vajamineva info (näiteks temperatuuri andmete asukohad selle päeva jaoks või väljundi kõvakettale kirjutamise asukoht).

Äramärkimist väärib meetod *run()*, mis arvutab vajaminevad väljundid (defineeritud kasutaja konfiguratsioonifailis) määratud ajaperioodi jaoks ja kirjutab kõvakettale soovitud vahe- ja lõpptulemused. Põhimõtteliselt võiks itereerida üle kuupäevade ja arvutada igaühe jaoks *GPP* kasutades meetodit *calc_gpp()* ning samuti kutsuda iga nõutud väljundi jaoks välja seda arvutav meetod, et selle tulemus siis kõvakettale kirjutada. Seda siiski tehtud pole. Nii siin kui ka klassis *NPP* on see meetod optimeeritud selliselt, et arvutus võtaks võimalikult vähe aega. Kuna sellise mudeli puhul on suurim ajakulu kõvakettalt lugemine ja sinna kirjutamine, siis on sedalaadi üleliigne tegevus sealt eemaldatud.

```

class GPP:
    def __init__(self, config):
        self.config = config
        self.land_cover = fetch_band(config.land_cover_path)
        self.temperature_stress_factor = TemperatureStressFactor(config)
        self.water_stress_factor = WaterStressFactor(config)

    def run(self)

    def write_dataset(self, data, out_dir, file_name, file_format=None, data_type=None, nodata_value=None)
    def write_daily_output(self, day, data, output_var, data_type='Float32')
    def write_monthly_output(self, day, data, output_var, data_type='Float32')
    def write_yearly_output(self, day, data, output_var, data_type='Float32')
    def calc_daily_temp(self, day, unit='K', column='temperature')
    def calc_daytime_cloud_cover(self, day, scale=1, column='cloud_cover')
    def calc_daytime_vpd(self, day, unit='Pa')
    def calc_par(self, day)

    def calc_temperature_stress_factor(self, day)
    def calc_water_stress_factor(self, day)
    def calc_gpp(self, day)

    def calc_gpp_for_land_cover_type(land_cover_code)

```

Joonis 10. GPP klassi muutujad ja meetodid. Tähtsaimad meetodid on punase piirjoonega esile tõstetud.

GPP arvutamise piirangufunktsioonid on implementeeritud moodulis eraldi klassidena nimedega *TemperatureStressFactor* ja *WaterStressFactor*, esimest neist on täpsemalt kirjeldatud ka joonisel 11. Ülesehituselt on need muidu sarnased.

```

class TemperatureStressFactor

    UNIVERSAL_TERRESTRIAL_ECOSYSTEM_MODEL = 'Universal terrestrial ecosystem model'
    TERRESTRIAL_ECOSYSTEM_MODEL = 'Terrestrial ecosystem model'
    MODIS = 'MODIS'

    def __init__(self, config):
        self.config = config
        self.land_cover = fetch_band(self.config.land_cover_path)
        self.function = TemperatureStressFactor.UNIVERSAL_TERRESTRIAL_ECOSYSTEM_MODEL

    def set_function(self, function)

    def calculate(self, day)

        def calc_tec_temp_stress_for_land_cover_type(land_cover_code)

        def calc_modis_temp_stress_for_land_cover_type(land_cover_code)

    def calc_daily_temp(self, day, unit='K', column='temperature')

```

Joonis 11. *TemperatureStressFactor* klassi muutujad ja meetodid. Tähtsaimad meetodid on punase piirjoonega esile tõstetud.

Lisaks eelpool kirjeldatud EST-PP mudeli temperatuuri piirangufunktsioonile, on EST-PP-Python'is võimalik kasutada ka kahte teist piirangufunktsiooni. Samuti on kergesti võimalik neid lisaarendusega lisada. Meetoditest on märkimist väärt *set_function()*, millega saab määrata, millist piirangufunktsiooni piiranguteguri arvutamiseks kasutada ja *calculate()*, mis selle arvutuse ära teeb.

3.6.4. npp.py

Moodul *npp.py* defineerib endas vaid ühe klassi: *NPP*. Sarnane klassiga *GPP* selle poolest, et ka siin on enamus meetodeid kas *NPP* arvutamiseks vajaminevate suuruste arvutamiseks või väljundite kõvakettale kirjutamiseks. Meetodeid on omajagu rohkem, sest *NPP* leidmiseks peab arvutama mitmeid hingamise tegureid, millede endi arvutamiseks peab eelnevalt arvutama vastavate taimekomponentide massi. Klassi *NPP* kirjeldab ka joonis 12. Võrreldes klassi *GPP* meetoditega on üks erinevus see, et nüüd on osade meetodite peamiseks argumendiks mitte kuupäev vaid aasta, tulenevalt eelpool kirjeldatud algoritmi loogikast. Meetod *calc_psnet()* arvutab MODIS GPP/NPP produktis leiduvat päevast suurus PsNet, mis kujutab endast päevast brutoproduksiooni, millest on maha lahutatud vastava päeva lehtede ja juurte raku hingamine.

```
class NPP
def __init__(self, config):
    self.config = config
    self.land_cover = fetch_band(self.config.land_cover_path)

def run(self)

def write_dataset(self, data, out_dir, file_name, file_format=None, data_type=None, nodata_value=None)
def write_daily_output(self, day, data, output_var, data_type='Float32')
def write_monthly_output(self, day, data, output_var, data_type='Float32')
def write_yearly_output(self, day, data, output_var, data_type='Float32')
def fetch_daily_output(self, day, output_var)
def calc_daily_temp(self, day, unit='K', column='temperature')
def calc_leaf_mass(self, day)    def calc_leaf_mass_for_land_cover_type(land_cover_code)
def calc_leaf_mr(self, day)    def calc_leaf_mr_for_land_cover_type(land_cover_code)
def calc_froot_mass(self, day)  def calc_froot_mass_for_land_cover_type(land_cover_code)
def calc_froot_mr(self, day)    def calc_froot_mr_for_land_cover_type(land_cover_code)

def calc_psnet(self, day, gpp=None)
def calc_leaf_mass_annual_max(self, year)
def calc_livewood_mass(self, year)    def calc_livewood_mass_for_land_cover_type(land_cover_code)
def calc_livewood_mr(self, day, livewood_mass=None)    def calc_livewood_mr_for_land_cover_type(land_cover_code)
def calc_annual_livewood_mr(self, year, livewood_mass=None)
def calc_annual_npp(self, year)
```

Joonis 12. *NPP* klassi muutujad ja meetodid. Tähtsaimad meetodid on punase piirjoonega esile tõstetud.

3.6.5. outputs_processing.py

Moodul *outputs_processing.py* sisaldab endas klassi *Outputs* ja selle meetodeid. Kui eelmised moodulid ja nende sisu tegelesid mudeli jooksumise, tulemuste arvutamisega ja rasterkujul kõvakettale kirjutamisega, siis *Outputs* meetodid töötavad nende tulemuste rasterfailidest koosnevate andmekomplektide peal ja töötlevad need ümber aegridade analüüsimiseks sobivasse formaati. Sisemiselt loetakse andmed rastritest *Pandas Dataframe*-i, kus ridadeks (*Pandase* terminoloogias indeksiks) on kuupäevad ja tulbad kujutavad endas mingit suurust. Klassi *Outputs* ja selle meetodeid kirjeldab joonis 13.

```
class Outputs
def __init__(self, config):
    self.config = config
    self.land_cover = fetch_band(config.land_cover_path)
def run(self)
def create_daily_output_pixels_time_series_dataframe(self, output_var, start_date=None, end_date=None, mask=None)
def create_daily_outputs_daily_spatial_averages_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
def create_daily_outputs_daily_spatial_sums_time_series_dataframe(self, output_vars, start_date=None, end_date=None,
    mask=None):
def create_daily_outputs_monthly_spatial_averages_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
def create_daily_outputs_monthly_spatial_sums_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
def create_daily_outputs_yearly_spatial_averages_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
def create_daily_outputs_yearly_spatial_sums_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
def create_daily_outputs_daily_spatial_stats_time_series_dataframe(self, output_vars, start_date=None,
    end_date=None, mask=None):
    stats_list = ['count', 'zeros', 'min', 'Q1', 'median', 'Q3', 'max', 'mean', 'sum', 'var', 'std', 'skew',
        'kurt']
def create_daily_outputs_daily_spatial_stats_per_land_cover_type_time_series_dataframe(self, output_vars,
    start_date=None,
    end_date=None, mask=None):
@staticmethod
def write_daily_outputs_daily_spatial_stats_per_land_cover_type_time_series_dataframe_to_csv(df, out_csv_file):
@staticmethod
def read_daily_outputs_daily_spatial_stats_per_land_cover_type_time_series_csv_to_dataframe(in_csv_file):
```

Joonis 13. *Outputs* klassi muutujad ja meetodid. Tähtsaimad meetodid on punase piirjoonega esile tõstetud.

Näiteks meetod *create_daily_output_pixels_time_series_dataframe()* loob ükskõik millise päevase väljundi jaoks kõigi mudeli võrgusilmade aegread ehk siis iga tulp kujutab endast ühe konkreetse mudeli võrgusilma päritud päevase väljundi aegrida. Meetod *create_daily_outputs_daily_spatial_stats_time_series_dataframe()* loob erinevate ruumiliste statistikute päevased aegread. Ala, mille üle statistikud arvutatakse, on ühe meetodi

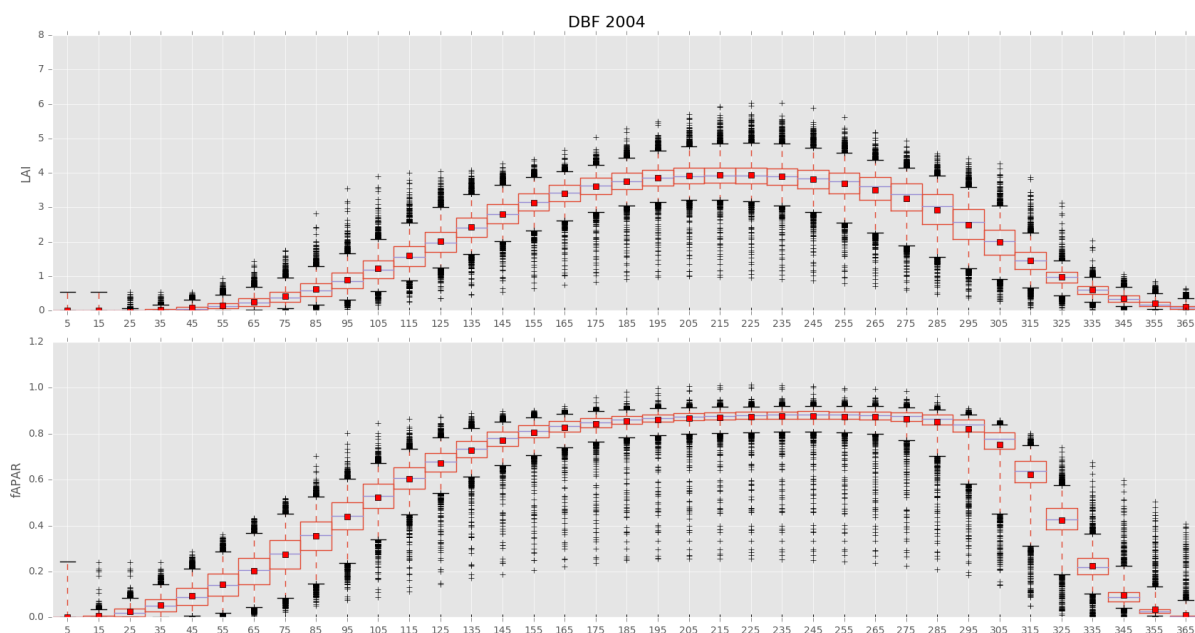
argumendina ette antav. Seda meetodit kasutab ära ka meetod `create_daily_outputs_stats_per_land_cover_type_time_sreis_dataframe()`, mis arvutab erinevad ruumilised statistikud maakattetüüpide kaupa iga päeva jaoks. Põhimõtteliselt on tegemist tulemuste kiiret analüüsi võimaldavate meetoditega, mida kasutades on lihtne erinevaid aegride graafikuid koostada. Samuti on olemas loodavate *Pandas DataFrame*-de csv formaadis kõvakettale kirjutamise meetod.

3.6.6. utils.py

Sisaldab mõnda kasulikku ja tihti kasutatavat funktsiooni, mis ei vaja töötamiseks ligipääsu detailsele jooksukonfiguratsioonile.

3.6.7. outputs_analysis.py

Skript `outputs_analysis.py`, mis ei ole põhikoodi osa vaid asub kaustas *scripts*, näitlikustab tulemuste analüüsimise võimalusi ja graafikute loomist mooduli `outputs_analysis.py` klassi *Outputs* meetoditega. Näitlikustav skript on eelkõige kasulik selleks, et uuele kasutajale tutvustada *Pandas DataFrame*-dega töötamist. Seda ka seetõttu, et osad *Outputs* meetodid loovad keskmisest natuke keerukama ülesehitusega *Pandas* andmestruktuure ja nendest soovitava info või andmeveeru pärimiseks peab hästi teadma *Pandas*-e funktsionaalsust ja andmestruktuuride ülesehitust. Allpool on joonisel 14 ära toodud graafikud, mis on toodetud *Outputs* klassi meetodite ja `outputs_analysis.py`-le sarnase skriptiga



Joonis 14. EST-PP-Python *Outputs* klassi meetodite ja `outputs_analysis.py`-le sarnase skriptiga loodud 2004 aasta lehtmetsade fAPAR ja LAI aegread.

3.7. Arenduse seis ja võimalikud edasiarenduse suunad

Praeguses seisus on arendust testitud vaid EST-PP sisendandmete peal ja operatsioonisüsteemis Windows. Mudelit on testitud ja kontrollitud, et ta väljastab korrektseid ja kvaliteetseid tulemusi, primaarproduktiooni mudeli algoritmilises vaates. Tarkvaraarenduslikus vaates on tegemist Beeta staadiumis projektiga, see tähendab, et plaanitud funktsionaalsused on implementeeritud, aga ilmselt eksisteerib varjatud vigasid. Samuti pole kirjutatud tarkvara testimise skripte, mis on klassikalise tarkvaraarenduse lahutamatu osa. Vajakajäämisi on kindlasti ka dokumentatsioonis, mis praegusel kujul piirdub käesoleva tööga. Seega võimalikud edasiarendused või muud tööd tarkvaraga on:

- 1) Luua tarkvara kirjeldav detailne dokumentatsioon
- 2) Luua testid, mis kontrolliksid tarkvara korrektset töötamist tarkvaralisest vaatest
- 3) Testida mudeli töötamist teiste sisendandmekomplektide peal
- 4) Täiendada mudelit võimalike erinevate piirangufunktsioonidega

3.8. Arutelu

Üldiselt võib öelda, et arendusele seatud eesmärgid said täidetud. Kui jätta välja kasutajapoolse manuaalse konfiguratsioonifaili täitmine, on terve töötlusprotsess algusest lõpuni automaatne ja ei nõua kasutajapoolset sekkumist. Ka selle käivitamine ja eelnev Python paketi paigaldamine on lahendatud standardsete levinud lahendustega, mis ei tohiks kasutajatele suuremaid probleeme tekitada. Konfiguratsioonifaili täitmist saab teha tavalise tekstiredaktoriga. Küll vajaks ilmselt lisamist dokumentatsioon, mis kirjeldaks täpselt ära kõik konfiguratsioonifailis määratavad suurused. Seda toetab tegelikult ka konfiguratsioonifaili failiformaadiks valitud YAML, mis võimaldab sellist infot enda sisus kajastada. Seega, kuigi selle implementeerimiseni ei jõutud, sai failiformaat valitud selliselt, et see toetab selle rakendamist ilma lisafaile tootmata.

Läbi üldise konfiguratsioonifaili on muudetavad ka kõik eesmärkides välja toodud jooksu või mudeli omadused (sisendandmestikud, piirangufunktsioonid, soovitud väljundid, modelleerimisala ja -ajaperiood), kuigi tõsi, nende kõigi töötamist põhjalikult testitud ei ole. Sellist paindlikku konfiguratsiooni võimaldab *config.py* mooduli *Config* klass, millest tuleb iga jooksu jaoks luua üks isend, mille muutujad täidetakse vastavalt konfiguratsioonifaili väärtustele. See isend antakse edaspidi kõigile töötlust tegevatele klassidele ette, kust töötluses vajaminevaid suurusi pärida. Sellise ülesehituse hea omadus on ka see, et iga jooksu üldine konfiguratsioonifail kirjeldab ära selle jooksu keskkonna (sisendandmete asukohad, loodud väljundid ja nende asukohad). See võimaldab Python'i arenduskeskkonnas, kasutades *estpp*

klasse ja nende meetodeid, analüüse ja lisatöötlust sisend- ja väljundandmete pealt teha ka pärast jooksu lõppemist. Selleks tuleb lihtsalt luua uuesti detailne jookskonfiguratsioon üldisest konfiguratsioonifailist (mille saab valida ühe väljundina jooksu väljundite kausta kirjutamiseks) kasutades *Config.create_detailed_config()* meetodit, see ette anda mõnele klassile, näiteks *Outputs* moodulist *outputs_processing.py*, mille meetodeid saab siis juba kasutada tulemuste edasiseks analüüsiks. Selline ülesehitus võimaldab ka tarkvara ja mudeli funktsionaalsust suhteliselt lihtsasti laiendada, sest peale *config.py* mooduli, ei ole teistes moodulites praktiliselt üldse sisse kirjutatud muutmatuid parameetreid, sest need päritakse kõik vastava jooksu *Config* isendilt. Sellised universaalselt kirjutatud klassid ja nende meetodid tagavad ka selle, et neid saab kasutada API stiilis enda töötlusprotsessi üles seadmiseks ja nendesse saab suhteliselt lihtsalt analoogseid universaalseid, aga uute funktsionaalsustega, meetodeid lisada.

Seega said kõik arendusele seatud eesmärgid täidetud.

Kokkuvõte

Käesolev magistritöö käsitleb maismaa taimkatte primaarproduktiooni ning selle globaalseks ja regionaalseks hindamiseks kasutatavaid kiirguse kasutamise efektiivsuse (KKE) põhimõtetel töötavaid mudeleid. Primaarproduktioon on orgaaniliste ühendite loomine atmosfääris leiduvast süsihappegaasist, üldjuhul fotosünteesi, aga ka kemosünteesi läbi. Tal on tähtis koht kiires süsinikuringes ja ka majanduslikult metsade või põllukultuuride kasvu hindamises. KKE põhimõte ütleb, et mingi teatud taimekoosluse jaoks eksisteerib maksimaalne päikesekiirguse abil süsiniku tootmise efektiivsus, mida piiravad erinevad meteoroloogilised ja keskkonnast olenevad tegurid. Sellist tüüpi mudelid on viimase kümnendi jooksul populaarsust kasvatanud, sest need on ülesehituselt suhteliselt lihtsad, eksisteerivad vajalike sisendite globaalsed andmekogud ja oma arvutusressursi vajadustest lähtuvalt on neid võimalik operatiivselt jooksutada.

Peamised erinevused sellist tüüpi mudelite vahel on nende maksimaalset kiirguse kasutamise efektiivsust vähendavate piirangufunktsioonide definitsioonides. Need võivad erineda kasutatavate sisendsuuruste poolest, aga ka piirangufunktsioonide üldiste kujude ja nende kombineerimise poolest. Samuti leidub mitmeid erinevaid mooduseid neelatud ja pealelangenu fototsünteesiliselt aktiivse kiirguse suhte leidmiseks. Erinevate samat meteoroloogilist või keskkonnast tulenevat suurust kirjeldavate andmekomplektide kasutamine ei oma niivõrd suurt mõju, sest kõik need mudelid sisaldavad ka referentsandmete abil leitavaid parameetreid, et mudelid kalibreerida.

Ühe sellist tüüpi mudeli rakendus Eesti ala peal on EST-PP ja selle rakendus Python'is EST-PP-Python. EST-PP on algoritmiliselt sarnane MODIS primaarproduktiooni produktiga, mis on üks enim analüüsitud seda tüüpi andmekiht. EST-PP-Python sai arendatud, sest esialgne EST-PP oli oma ülesehituselt erinevate individuaalsete skriptide kogum, mis ei võimaldanud teda mugavalt ja efektiivselt jooksutada. Ka sai EST-PP-Python's implementeeritud võimalus erinevaid piirangufunktsioone kasutada, sisendandmeid ja modelleerimisala muuta ning palju muud. Mudelit saab jooksutada käsurealt, samuti on võimalik ta Python'is pakatina importida ja selle klasse ning meetodeid kasutada enda töötlusprotsessi loomiseks. Põhimõtteliselt sai loodud palju universaalsem, kergelt konfigureeritav palju kasutajasõbralikum EST-PP mudeli rakendus, mida on võimalik kasutada edasiseks teadustöök, näiteks Eesti metsade juurdekasvu hindamiseks, või õppetöös.

Kasutatud kirjandus

- [1] J. Dong, X. Xiao, P. Wagle, G. Zhang, Y. Zhou, C. Jin, M. S. Torn, T. P. Meyers, A. E. Suyker, J. Wang, H. Yan, C. Biradar ja B. Moore III, „Comparison of four EVI-based models for estimating gross primary production of maize and soybean croplands and tallgrass prairie under severe drought,“ *Remote Sensing of Environment*, kd. 162, pp. 154-168, 2015.
- [2] L.-X. Zhang, D.-C. Zhou, J.-W. Fan ja Z.-M. Hu, „Comparison of four light use efficiency models for estimating terrestrial gross primary production,“ *Ecological Modelling*, kd. 300, pp. 30-39, 2015.
- [3] C. Wu, W. J. Munger, Z. Niu ja D. Kuang, „Comparison of multiple models for estimating gross primary production using MODIS and eddy covariance data in Harvard Forest,“ *Remote Sensing of Environment*, kd. 114, pp. 2925-2939, 2010.
- [4] N. C. Coops, „Characterizing Forest Growth and Productivity Using Remotely Sensed Data,“ *Current Forestry Reports*, pp. 195-205, 2015.
- [5] S. W. Running ja M. Zhao, „Daily GPP and Annual NPP (MOD17A2/A3) Products NASA Earth Observing System MODIS Land Algorithm,“ 2015.
- [6] W. Yuan, S. Liu, G. Zhou, G. Zhou, L. L. Tieszen, D. Baldocchi, C. Bernhofer, H. Ghloz, A. H. Goldstein, M. L. Goulden, D. Y. Hollinger, Y. Hu, B. E. Law, P. C. Stoy, T. Vesala ja S. C. Wofsy, „Deriving a light use efficiency model from eddy covariance flux data for predicting daily gross primary production across biomes,“ *Agricultural and Forest Meteorology*, kd. 143, pp. 189-207, 2007.
- [7] M. He, W. Ju, Y. Zhou, J. Chen, H. He, S. Wang, H. Wang, D. Guan, J. Yan, Y. Li, Y. Hao ja F. Zhao, „Development of a two-leaf light use efficiency model for improving the calculation of terrestrial gross primary productivity,“ *Agricultural and Forest Meteorology*, kd. 173, pp. 28-39, 2013.
- [8] H. Yan, S.-q. Wang, D. Billesbach, W. Oechel, G. Bohrer, T. Meyers, T. A. Martin, R. Matamala, R. P. Phillips, F. Rahman, Q. Yu ja H. H. Shugart, „Improved global

- simulations of gross primary product based on a new definition of water stress and a separate treatment of C3 and C4 plants," *Ecological Modelling*, kd. 297, pp. 42-59, 2015.
- [9] I. McCallum, O. Franklin, E. Moltchanova, L. Merbold, C. Schmullius, A. Shvidenko, D. Schepaschenko ja S. Fritz, „Improved light and temperature responses for light-use-efficiency-based GPP models," *Biogeosciences*, kd. 10, pp. 6577-6590, 2013.
- [10] S. Wang, K. Huang, H. Yan, H. Yan, L. Zhou, H. Wang, J. Zhang, J. Yan, L. Zhao, Y. Wang, P. Shi, F. Zhao ja L. Sun, „Improving the light use efficiency model for simulating terrestrial vegetation gross primary production by the inclusion of diffuse radiation across ecosystems in China," *Ecological Complexity*, kd. 23, pp. 1-13, 2015.
- [11] W. Yuan, W. Cai, J. Xia, J. Chen, S. Liu, W. Dong, L. Merbold, B. Law, A. Afrain, J. Beringer, C. Bernhofer, A. Black, P. D. Blanken, A. Cescatti, Y. Chen, L. Francois, D. Gianelle, I. A. Janssens, M. Jung, T. Kato, G. Kiely, D. Liu, B. Marcolla, L. Montagnani, A. Raschi, O. Roupsard, A. Varlagin ja G. Wohlfahrt, „Global comparison of light use efficiency models for simulating terrestrial vegetation gross primary production based on the LaThuile database," *Agricultural and Forest Meteorology*, Kd-d. %1/%2192-193, pp. 108-120, 2014.
- [12] W. Cai, W. Yuan, S. Liang, S. Liu, W. Dong, Y. Chen, D. Liu ja H. Zhang, „Large Differences in Terrestrial Vegetation Production Derived from Satellite-Based Light Use Efficiency Models," *Remote Sensing*, pp. 8945-8965, 2014.
- [13] T. Nilson, M. Rennel, A. Luhamaa, M. Hordo, A. Olesk ja M. Lang, „MERIS GPP/NPP product for Estonia: I. Algorithm and preliminary results of simulation," *Forestry Studies*, pp. 56-78, 2012.
- [14] „Nutrient Cycles in Ecosystems," [Võrgumaterjal]. Available: <http://moodle.sd74.bc.ca/mod/book/view.php?id=690&chapterid=837>.
- [15] NASA, „The Fast Carbon Cycle," [Võrgumaterjal]. Available: <https://earthobservatory.nasa.gov/Features/CarbonCycle/page3.php>.
- [16] „Primaarproduktsioon," Vikipeedia, [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Primaarproduktsioon>.

- [17] A. Ito ja T. Oikawa, „Global Mapping of Terrestrial Primary Productivity and Light-Use Efficiency with a Process-Based Model,“ *Global Environmental Change in the Ocean and on Land*, pp. 343-358, 2004.
- [18] J. L. Monteith, „Solar Radiation and Productivity in Tropical Ecosystems,“ 1972.
- [19] J. L. Monteith ja C. J. Moss, „Climate and the Efficiency of Crop Production in Britain,“ *Philosophical Transactions of the Royal Society*, 1977.
- [20] M. He, Y. Zhou, W. Ju, J. Chen, L. Zhang, S. Wang, N. Saigusa, R. Hirata, S. Murayama ja Y. Liu, „Evaluation and improvement of MODIS gross primary productivity in typical forest ecosystems of East Asia based on eddy covariance measurements,“ *Journal of Forest Research*, kd. 18, pp. 31-40, 2013.
- [21] P. E. Thornton, B. E. Law, L. Gholz, K. L. Clark, E. Falge, D. S. Ellsworth, A. H. Goldstein, R. K. Monson, D. Hollinger, M. Falk, J. Chen ja J. P. Sparks, „Modeling and measuring the effects of disturbance history and climate carbon and water budgets in evergreen needleleaf forests,“ *Agricultural and Forest Meteorology*, kd. 113, pp. 185-222, 2002.
- [22] E. E. Online, „MERIS,“ [Võrgumaterjal]. Available: <https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/envisat/instruments/meris>.
- [23] A. Luhamaa, K. Kimmel, A. Männik ja R. Rõõm, „High resolution re-analysis for the Baltic Sea region during 1965-2005,“ *Climate Dynamics*, kd. 36, pp. 727-738, 2011.
- [24] „HIRLAM,“ [Võrgumaterjal]. Available: <http://hirlam.org/>.
- [25] NASA, „Global Modeling and Assimilation Office,“ [Võrgumaterjal]. Available: <https://gmao.gsfc.nasa.gov/>.
- [26] A. Strahler, D. Muchoney, J. Borak, M. Friedl, S. Gopal, E. Lambin ja A. Moody, „MODIS Land Cover and Land-Cover Change,“ 1999.
- [27] M. Friedl, „User Guide for the MODIS Land Cover Type Product (MCD12Q1),“ 2012.
- [28] JetBrains, „PyCharm. Python IDE for Professional Developers,“ [Võrgumaterjal]. Available: <https://www.jetbrains.com/pycharm/?fromMenu>.

- [29] Atlassian, „Code, Manage, Collaborate. Bitbucket is the Git solution for professional teams,“ [Võrgumaterjal]. Available: <https://bitbucket.org/product>.
- [30] „git --local-branching-on-the-cheap,“ [Võrgumaterjal]. Available: <https://git-scm.com/>.
- [31] Atlassian, „Sourcetree,“ [Võrgumaterjal]. Available: <https://www.sourcetreeapp.com/>.
- [32] Atlassian, „Atlassian,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/>.

Lisad

Lisa 1. user_config.yaml

```
!!omap
- inputs:
  PAR:
  BPLUT:
  inDir:
  land_cover:
  inDirs:
    temperature:
    LAI:
    relative_humidity:
    cloud_cover:
    fAPAR:
    sw_radiation:
- run_dir:
- run_name:
- time_period:
- temp_stress_function: Universal terrestrial ecosystem model
- modelling_area:
  params:
    proj_WKT:
    geotransform:
    proj_EPSG:
    nx:
    ny:
    from_file: E:\EST-PP\EST-PP\Input\modelling_grid.vrt
- water_stress_function: MODIS
- runs_dir:
- output:
  complex_meteo_factor: false
  PAR: true
  froot_MR: false
  livewood_mass: false
  temp_stress_factor: true
  BPLUT: true
  water_stress_factor: true
  PSNet: true
  GPP: true
  vpd_avg: false
  NPP: true
  cloud_avg: false
  fAPAR: false
  APAR: false
  LAI: false
  log_file: true
  leaf_MR: false
  temp_avg: false
  froot_mass: false
  livewood_MR: false
  leaf_mass: false
  config: true
```


Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, _____ Mattias Rennel _____,

(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Kiirguse kasutamise efektiivsuse põhimõttel töötava maismaa taimkatte primaarproduktiooni
mudeli _____ rakendus _____ Python'is.

(*lõputöö pealkiri*)

mille juhendaja on _____ Piia Post _____,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **29.05.2017**